

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроніки

(повна назва інституту/факультету)

Кафедра звукотехніки та реєстрації інформації

(повна назва кафедри)

«На правах рукопису»

УДК 778.534.9

«До захисту допущено»

Завідувач кафедри

Г.Г.Власюк

(підпис)

(ініціали, прізвище)

“ ” _____ 2019 р.

Магістерська дисертація

спеціальність _____ 171 Електроніка _____
(код і назва спеціальності)

на тему: «Побутова система для контрольованого нагрівання води,
як елемент Інтернету речей»

Виконав: студент VI курсу, групи ДВ-81мп _____
(шифр групи)

Колтунов Кирило Дмитрович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник професор, к.т.н., професор Пілінський В.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Інститут (факультет) Факультет електроніки
(повна назва)

Кафедра звукотехніки та реєстрації інформації
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (освітня програма) 171 Електроніка

(Електронні системи мультимедіа та засоби Інтернету речей)

(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Г.Г.Власюк
(підпис) (ініціали, прізвище)

« » 20 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Колтунову Кирилу Дмитровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Побутова система для контрольованого нагрівання води,
як елемент Інтернету речей»

науковий керівник дисертації Пілінський Володимир Володимирович, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 07 » листопада 2019 р. № 3859-с

2. Строк подання студентом дисертації 09.12.2019 р.

3. Об'єкт дослідження: система контрольованого нагрівання води

4. Предмет дослідження (Вхідні дані – для магістерської дисертації за спеціальністю 171 Електроніка): Мікроконтролери Arduino Uno – 16 МГц, датчик температури DS18B20: від 1°C до 125°C, датчик тиску MPXV5050: від 20 до 200 через АЦП, інфрачервоний модуль: 20 метрів, водяна помпа: 5мл/с, модуль Bluetooth: 25 метрів, App Inventor: 4 Мб.

5. Перелік завдань, які потрібно розробити: 1) Визначити передумови використання засобів мікропроцесорної техніки для пристроїв контрольованого нагрівання води; 2) Провести теоретичний аналіз методу вимірювання рівня води; 3) Запропонувати рішення по вдосконаленню мікроконтролера для збільшення входів та виходів; 4) Виконати реалізацію пристрою для контрольованого нагрівання води засобами мікропроцесорної техніки.

6. Перелік графічного (ілюстративного) матеріалу: 1) 78 рис, 23 табл., 1 презентація, 8 слайдів.

7. Орієнтовний перелік публікацій: 1.) Цифрове вимірювання рівня гарячої води на мікропроцесорній платформі Arduino // International Scientific And Practical Conference «Technical Sciences: History, The Present Time, The Future, EU Experience», 2019.,-С.69-72 2.) Усунення брязкоту контактів під час комутації сигналу 3.) Інтерпретація аналого-цифровим перетворювачем мікроконтролера індивідуальної напруги для створення декількох цифрових входів на одному аналоговому// Slovak International Scientific Journal. №35, 2019.- С. 24-30.

8. Дата видачі завдання 10. 09. 2018 р.

Календарний план

№ з / п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Написання першого розділу: Передумови використання засобів мікропроцесорної техніки для системи контрольованого нагрівання води	15.12.2018	виконано
2	Написання другого розділу: Теоретичні та практичні засади розробки пристрою	30.05.2019	виконано
3	Написання третього розділу: Реалізація пристрою для контрольованого нагрівання води, як елементу інтернету речей	10.10.2019	виконано
4	Підготовка матеріалів до друку та оформлення пояснювальної записки	30.11.2019	виконано
5	Підготовка та оформлення презентації для доповіді	03.12.2019	виконано

Студент

(підпис) К.Д. Колтунов
(ініціали, прізвище)

Науковий керівник

(підпис) В.В. Пілінський
(ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 137 с., 78 рис., 23 табл., 2 дод., 52 джерела,
ВИМІРЮВАННЯ РІВНЯ ВОДИ, ВИМІРЮВАННЯ ТЕМПЕРАТУРИ,
МІКРОКОНТРОЛЕР, МІКРОПРОЦЕСОРНА ТЕХНІКА, APP INVENTOR 2,
ARDUINO, BLUETOOTH

Актуальність теми роботи полягає в тому, що сучасні компанії, які виробляють побудові електричні обладнання прагнуть до зменшення цін на системи нагрівання води, і як наслідок використовують примітивні технології, не використовуючи засоби мікропроцесорної техніки.

Об'єктом дослідження є використання засобів мікропроцесорної техніки для створення пристрою контрольованого нагрівання води.

Метою дослідження є створення системи контрольованого нагрівання води внаслідок використання засобів мікропроцесорної техніки. Визначити найкращі варіанти реалізації в плані функціональності, вартості пристрою та загального результату для подальшого застосування в повсякденному житті.

Для досягнення поставленої мети необхідно виконати такі завдання:

- Опрацювати існуючі способи дослідження та моделювання систем для контрольованого нагрівання води.
- Створити систему для контрольованого нагрівання води.
- Розробити програмне забезпечення для системи нагрівання води та керування його налаштувань.
- Побудувати макет системи та налаштувати усіх електричні компоненти.
- Розробити програмне забезпечення для керування макетом зі смартфона.

1.

SUMMARY

Master's dissertation: 131 p., 78 pic., 23 tabl., 52 sources, 2 supplements.

MICROPROCESSOR ENGINEERING, ARDUINO, MICROCONTROLLER, BLUETOOTH, MEASUREMENT OF WATER LEVEL, MEASUREMENT OF TEMPERATURE, APP INVENTOR 2

The relevance of the topic of the work lies in the fact that modern companies that manufacture electrical equipment tend to reduce prices for water heating systems, and consequently, use primitive technologies without using microprocessor technology.

The object of the study is the using of microprocessor technology to create a controlled water-heating device.

The purpose of the study is to create a system of controlled water heating by using microprocessor technology. Determine the best implementation options in terms of functionality, cost, and overall output for future using in everyday life.

To achieve the goal, you need to accomplish the following tasks:

- Develop existing methods of researching and modeling systems for controlled water-heating devices.
- Create a controlled water-heating system.
- Develop software for the water-heating system and for control its settings.
- Build a system layout and customize all electrical components.
- Develop mobile software for control system by smartphone.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	8
ВСТУП	9
1. ПЕРЕДУМОВИ ВИКОРИСТАННЯ ЗАСОБІВ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ ДЛЯ СИСТЕМИ КОНТРОЛЬОВАНОГО НАГРІВАННЯ ВОДИ.....	11
1.1.Розвиток мікропроцесорної техніки.....	12
1.2.Огляд мікроконтролерів	14
1.2.1. Arduino Uno.....	15
1.2.2. Arduino Mega.....	16
1.3. Огляд периферійних пристроїв.....	18
1.3.1. Пристрої вводу інформації.....	19
1.3.1.1. Потенціометр.....	19
1.3.1.2. Фоторезистор.....	20
1.3.1.3. Датчик температури DS18B20.....	24
1.3.1.4. Датчик тиску MPXV5050.....	28
1.3.1.5. Інфрачервоний модуль.....	32
1.3.2. Пристрої виводу інформації.....	34
1.3.2.1. Реле для комутації.....	33
1.3.2.2. Драйвер двигуна.....	37
1.3.2.3. Сервопривід.....	41
1.3.2.4. Водяна помпа.....	44
1.3.2.5. RGB – світлодіод.....	45
1.3.2.6. Модуль звуку.....	47
1.3.2.7. Дисплей LCD 2004.....	53
1.3.3. Пристрій вводу та виводу інформації - модуль Bluetooth.....	56
2. ТЕОРЕТИЧНІ ТА ПРАКТИЧНІ ЗАСАДИ РОЗРОБКИ ПРИСТРОЮ.....	59
2.1. Додаткові бібліотеки для розробки коду	59
2.1.1. LiquidCrystal_I2C.h.....	59
2.1.2. OneWire.h.....	60
2.1.3. ServoSmooth.h.....	63
2.1.4. GyverButton.h.....	64

	7
2.2. Види інтерфейсів для взаємодії периферійних пристроїв.....	66
2.2.1. SPI.....	67
2.2.2. I2C.....	68
2.2.3. UART.....	70
2.3. Вдосконалення мікроконтролера Arduino Uno.....	72
2.3.1. Брязкіт контактів - програмне і апаратне усунення.....	69
2.3.2. Підключення декілька кнопок до одного аналогового входу.....	71
2.3.3. Збільшення аналогових входів завдяки мультиплексору CD4051.....	80
2.3.4. Збільшення цифрових виходів через зсувний регістр 74НС595.....	83
3. РЕАЛІЗАЦІЯ ПРИСТРОЮ ДЛЯ КОНТРОЛЬОВАНОГО НАГРІВАННЯ ВОДИ, ЯК ЕЛЕМЕНТУ ІНТЕРНЕТУ РЕЧЕЙ.....	92
3.1. Конструювання пристрою контрольованого нагрівання води.....	92
3.1.1. Підключення електричних компонентів.....	93
3.1.2. Принципова схема пристрою.....	94
3.1.3. Макет для корпусу пристрою.....	95
3.1.4. Приклад програмної реалізації.....	96
3.1.5. Принцип дії пристрою.....	97
3.2. Розробка програмного середовища на смартфон для керування пристроєм.....	99
3.2.1. Середовище розробки App Inventor	100
3.2.2. Приклад програмної реалізації.....	101
3.2.3. Принцип дії програми.....	102
4. СТАРТАП ПРОЕКТ.....	103
4.1. Загальні відомості.....	103
4.2. Технологічний аудит ідеї проекту.....	103
4.3. Аналіз ринкових можливостей запуску стартап-проекту.....	104
4.4. Розроблення ринкової стратегії проекту	105
4.5. Розроблення маркетингової програми стартап-проекту.....	107
ВИСНОВКИ	113
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	114
Додаток А. Лістинг програми.....	118
Додаток Б. ABSTRACT.....	127

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЦП	–	Аналого-Цифровий Перетворювач;
ШИМ	–	Широтна-Імпульсна Модуляція;
IDE	–	Integrated Development Environment;
I2C	–	Inter-Integrated Circuit;
LCD	–	Liquid Crystal Display;
RGB	–	Red Green Blue;
SPI	–	Serial Peripheral Interface;
UART	–	Universal Asynchronous Receiver-Transmitter.

ВСТУП

Характерною ознакою ХХІ сторіччя є стрімкий розвиток систем інтернету-речей та автоматичних засобів керування приладів. Інтернет-речей набуло широкого розповсюдження у різних сферах життя, як на виробництві та корпораціях, так і у повсякденному житті. Це обумовлює розвиток технологій інтернету-речей. Побутова система для контрольованого нагрівання води – є новим та ще не розробленим пристроєм інтернету речей.

Сучасні компанії прагнуть до зменшення цін систем нагрівання води, тому використовують примітивні технології, не використовуючи засоби мікропроцесорної техніки. А ті хто, використовують та намагаються зробити пристрій автоматичного контролювання нагрівання води – завишають ціни на пристрій, так що вони становляться вже не «побутовими». Тому доцільно дослідити особливості реалізації пристроїв нагрівання води, з урахуванням різних наявних систем. Та запропонувати свою, яка має більший функціонал та меншу вартість реалізації.

Для досягнення поставленої мети необхідно: опрацювати існуючі способи дослідження та моделювання систем для контрольованого нагрівання води, створити систему для контрольованого нагрівання води, розробити програмне забезпечення для системи нагрівання води та керування його налаштувань, побудувати макет системи та налаштувати усіх електричні компоненти, розробити програмне забезпечення для керування макетом зі смартфона.

Таким чином запропоновано удосконалення технології, що забезпечують вимірювання рівня води та контроль температури у побутових пристроях нагрівання води, які побудовані на використанні мікропроцесорної техніки. Реалізовано систему, яка працює на базі мікроконтролера Arduino Uno. Мікроконтролер вдосконалено для підключення більшої кількості периферійних приладів та елементів комутацій сигналу, які потребує система для задуманої роботи пристрою. Система повністю автоматизована і може бути керованою, як органами управління, так і смартфоном на відстані.

Об'єкт дослідження – мікропроцесорна система контрольованого нагрівання води.

Предмет дослідження – використання засобів мікропроцесорної техніки для створення побутового пристрою контрольованого нагрівання води

Методи дослідження – теорія електричних кіл, основи мікроелектроніки, цифрова схемотехніка, проведення експерименту

Наукова новизна отриманих результатів: запропоновано удосконалення технології, що забезпечують вимірювання рівня води та контроль температури у побутових пристроях нагрівання води, які побудовані на використанні мікропроцесорної техніки.

Практична цінність отриманих результатів: реалізовано систему, яка працює на базі мікроконтролера Arduino Uno. Мікроконтролер вдосконалено для підключення більшої кількості периферійних приладів та елементів комутацій сигналу, які потребує система для задуманої роботи пристрою. Система повністю автоматизована і може бути керованою, як органами управління, так і смартфоном на відстані.

Апробація результатів роботи. Результати роботи опубліковані в міжнародних науково-технічних журналах та конференціях:

1. International Scientific And Practical Conference «Technical Sciences: History, The Present Time, The Future, EU Experience» (2019).
2. №35, 2019 Slovak International Scientific Journal.(2019).

Публікації

2. Цифрове вимірювання рівня гарячої води на мікропроцесорній платформі Arduino // International Scientific And Practical Conference с.59-62

3. Усунення брязкоту контактів під час комутації сигналу

Інтерпретація аналого-цифровим перетворювачем мікроконтролера індивідуальної напруги для створення декількох цифрових входів на одному аналоговому // Slovak International Scientific Journal №35

1 ПЕРЕДУМОВИ ВИКОРИСТАННЯ ЗАСОБІВ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ ДЛЯ СИСТЕМИ КОНТРОЛЬОВАНОГО НАГРІВАННЯ ВОДИ

Мікропроцесор - синхронізована цифрова інтегральна схема, яка приймає двійкові дані на вхід [1], обробляє їх відповідно до інструкцій, що зберігаються у його пам'яті, і забезпечує результати як вихід [2]. Мікропроцесори містять як комбінаційну логіку, так і послідовну цифрову логіку. Мікропроцесори працюють на цифрах і символах, представлених у системі двійкових чисел.

Інтеграція цілого процесора на єдиний чіп або декілька чіпів суттєво знизилася вартість технологічної потужності, підвищивши ефективність. Процесори інтегральної схеми виробляють в великій кількості за допомогою високо автоматизованих процесів, що призводить до низької вартості. Одночипові процесори підвищують надійність, тому що в них набагато менше електричних з'єднань для відмови. Оскільки мікропроцесорні конструкції покращуються, вартість виготовлення чіпа (з меншими компонентами, побудованими на такому ж розмірі напівпровідникового чіпа), як правило, залишається незмінною.

Перед мікропроцесорами розробляли невеликі комп'ютери з використанням плат, на які встановлювали багато середніх та малих інтегральних мікросхем. На заміну їм з'явилися мікроконтролери. Продовжується збільшення мікропроцесорної потужності через те, що інші форми комп'ютерів майже повністю застаріли. Пристрої з одним або кількома мікропроцесорами використовуються майже усюди, починаючи від найменших вбудованих систем і портативних пристроїв, до найбільших мікропроцесорів і суперкомп'ютерів.

1-1 Розвиток мікропроцесорної техніки

До розробки мікропроцесора існувало безліч ранніх технологій для моделювання логічних функцій в обчислювальних пристроях. Одні із першопрохідців у всесвіт мікропроцесорної техніки стала компанія IBM. Обчислювальні технології на базі вакуумних трубок і транзисторів вивели IBM на перші позиції у великій обчислювальній галузі, але ще не були призначені для бізнесу чи домашнього використання через їх надмірну вартість та інтенсивний графік технічного обслуговування. Ранні інтегральні схеми з'явилися у калькуляторах на початку 1960-х років, перш ніж Intel почав роботу над першим мікропроцесором [3].

1.1.1 4-бітні мікросхеми

Незважаючи на те, що IBM був титаном у ранніх масштабних обчислювальних змаганнях, компанія Intel яка заснована в 1968 році, забезпечила значну частину новаторської роботи, пов'язаної з технологією мікропроцесорів першого покоління. Потужність даного мікропроцесора все ще вимірювалась в бітах. Простіше кажучи, це стосується найбільш базової одиниці кодованих вказівок, виражених в рядку двійкових 1 та 0, які комп'ютер інтерпретує для виконання завдань. Чим потужніший процесор, тим більше інструкцій він може виконати одночасно, що призведе до швидшої обробки та більшої ефективності при складних завданнях

1.1.2 8-бітні мікросхеми

Мікропроцесори с 8-бітним чіпом не були дуже довго на ринку, тому найбільшим їх виробником стала компанія Intel, яка залишився попередником в ранніх технологіях мікропроцесорів, випустила свій перший 8-бітний

мікропроцесор – 8008, в 1972 році. Багато хто з тих же інженерів, які займалися оригінальними 4-бітними моделями, виробляли ідеї та технічні характеристики для цього нового покоління чіпів. До цього часу інші компанії почали свої розробки в області мікропроцесорів. Новий 8080-мікропроцесор Intel встановив основні напрямки комп'ютерних розробок і розпочав розвиток мікропроцесорів.

1.1.3 16-бітові мікросхеми

У середині 1970-х років National Semiconductor [4] потрапив у галузь розробки мікропроцесорів, через швидкі темпи прориву в Intel. 16-розрядні мікропроцесори, в тому числі National Semiconductor PACE, були відносно повільними, хоча з більш розвиненою ітерацією, такою як в Motorola 6800, з'явилися в ранніх системах Macintosh та інших. До того часу, коли 16-бітові мікросхеми досягли свого потенціалу, 32-розрядна ера вже почала приходити, та затьмарила 16-бітну технологію.

1.1.4 32-розрядні чіпи

До кінця 1970-х років розробка 32-розрядних мікропроцесорів проходила дуже активно, і вони почали з'являтися на масовому ринку у 1980-х роках. Виробниками стали такі компанії, як National Semiconductor і Hewlett-Packard [5]. Настільні інженерні пристрої вийшли на нову фазу з початком 32-розрядної обробки. 32-розрядні мікропроцесори були порівняно технологічні і можуть вважатися як попередник сучасних мікропроцесорів.

1.1.5 64-розрядні чіпи

64-розрядні чіпи стали доступні з 1992 року і розроблені в основному для комп'ютерів. Значну частину 64-бітної мікропроцесорної епохи домінувала битва між Intel та AMD. Остання компанія була заснована в 1969 році, і після того, як

підписала угоду з IBM та Intel, в них виросла популярність, завдяки чому AMD стає другим джерелом виробників Intel-розроблених процесорів. З цієї угоди почався розвиток енергійного ринку мікропроцесорів з двома великими компаніями та потенціалом для багатьох інших, оскільки технологія ще розвивається.

1.1.6 Чіпи RISC

Чіпи RISC з обмеженим набором інструкцій, походять від дослідницького проекту IBM, що проводився з другої половини 1970-х років [6]. Чіпи RISC були призначені для оптимізації комп'ютерного програмування, видаливши найскладніші інструкції та більш ефективно покладаючись на більш прості, більш загальні інструкції, які можна було б обробляти та виконувати швидше. Такі чіпи все ще використовуються сьогодні і перетворилися на спектр повноцінних комп'ютерних архітектур у конкурентній боротьбі з методами розробки CISC, які повною мірою застосовують складні інструкції. Архітектуру RISC широко використовують в таких пристроях, як мобільні телефони та певні типи комп'ютерів для подорожей.

1.2 Огляд мікроконтролерів

Сьогоднішні мікропроцесори є надзвичайно потужними, здатними виконувати складні інструкції швидше, ніж будь-коли раніше. На даний час є багато різноманітних мікроконтролерів, тому під час проектування цифрової системи необхідно здійснити його правильний вибір. Основна мета задовольнити специфікації системи та знизити загальну вартість системи. Для цього потрібно виконати вимоги з продуктивності, надійності, умов застосування та вартості. Основні критерії вибору мікроконтролера для системи, що досліджена у роботі є:

- придатність для прикладної системи;
- число контактів, портів введення-виведення;

- наявність периферійних пристроїв, таких як аналого-цифровий, цифро-аналоговий перетворювач та інтерфейс зв'язку;
- наявність інших периферійних пристроїв, які не будуть потрібні;
- можливість ядра мікроконтролера забезпечити необхідну продуктивність.

А також: доступність, підтримка розробника, наявність необхідного компілятора, надійність фірми виробника, компетентність, підтверджена розробками, надійність виробництва, тобто якість продукції, час роботи в цій галузі. Під зазначені критерії підходять мікроконтролери виробника Arduino, а саме моделі: Arduino Uno та Arduino Mega.

1.2.1 Arduino Uno

Arduino Uno - це плата мікроконтролерів з відкритим кодом на базі мікроконтролера ATmega328P, яка розроблена компанією Arduino.cc [7].



Рисунок 1.1 – Мікроконтролер Arduino Uno

Плата обладнана наборами цифрових та аналогових вводів та виводів, які можуть бути з'єднані з різними розширеннями. Плата має 14 цифрових штифтів і 6 аналогових штифтів. Програмується в Arduino IDE (інтегроване середовище розробки) із застосуванням USB-кабелю типу B. Він може працювати на USB-кабелі або на зовнішньому 9-вольтовому акумуляторі, хоча він приймає напругу

від 7 до 20 вольт. Він схожий на Arduino Nano і Leonardo. Апаратний еталонний дизайн поширюється за ліцензією Creative Commons Attribution Share-Alike 2.5 і доступний на веб-сайті Arduino. Макети та файли для виробництва для деяких версій апаратного забезпечення також є у вільному доступі. "Uno" означає «один» з італійської і було обрано для позначення випуску Arduino Software (IDE) 1.0. Плата Uno та версія 1.0 Arduino Software (IDE) були еталонними версіями Arduino, тепер вони розвивалися до нових версій [8]. Плата Uno - перша в серії плат USB Arduino та еталонна модель для платформ Arduino. ATmega328 на Arduino Uno попередньо запрограмована завантажувачем, який забезпечує завантаження нового коду без використання зовнішнього програмного забезпечення. Вона працює, використовуючи оригінальний протокол STK500 [9]. Uno також відрізняється від усіх попередніх плат, оскільки вона не використовує чіп FTDI USB-to-serial driver драйвер. Замість цього, він оснащений Atmega16U2, запрограмований як перетворювач USB-to-serial. Arduino Uno, як правило, вважають найбільш зручною та популярною платою на мікропроцесорній техніці, за порівнянням з платами, що продають менш ніж 5 доларів США.

Проект Arduino розпочався в Інституті дизайну взаємодії Ivrea (IDII) в Івреї, Італія. У той час студенти використовували мікроконтролер BASIC Stamp вартістю 100 доларів США, що є значними витратами для багатьох студентів. Але в 2003 році Ернандо Барраган створив платформу для розробки Wiring в якості магістерського дипломного проекту IDII, під керівництвом Массімо Банзі та Кейсі Реас. Ця платформа й стала платою Arduino Uno.

Технічні характеристики плати Arduino Uno:

- мікроконтролер: ATmega328P;
- робоча напруга: 5 В;
- вхідна напруга: (7...20) В;
- цифрові вхідні виходи: 14 (з яких 6 забезпечують ШІМ-вихід);
- аналогові вхідні штирі: 6;
- сила постійного струму на виводі: 20 мА
- струм постійного струму для контактів 3.3 В: 50 мА

- флеш-пам'ять: 32 кБ, з яких 0,5 кБ використовує завантажувач;
- SRAM: 2 кБ;
- EEPROM: 1 кБ;
- тактова частота 16 МГц;
- довжина: 68,6 мм;
- ширина: 53,4 мм;
- маса: 25 г.

1.2.2 Arduino Mega 2560

Arduino Mega 2560 - це пристрій на основі мікроконтролера ATmega2560 [10]. У його склад входить все необхідне для зручної роботи з мікроконтролером: 54 цифрових входів та виводів (з яких 15 можуть використовуватися в якості ШІМ-виходів), 16 аналогових входів, 4 UART (апаратних приймача для реалізації послідовних інтерфейсів), кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм ICSP для програмування і кнопка скидання. Для роботи з пристроєм треба подати живлення від AC/DC-адаптера або підключити його до комп'ютера USB-кабелем. Arduino Mega сумісна з більшістю плат розширення, розроблених для Arduino Duemilanove і Diecimila.



Рисунок 1.2 - Мікроконтролер Arduino Mega 2560

Ардуіно Mega 2560 відрізняється від усіх попередніх плат тим, що в ньому для перетворення інтерфейсів USB-UART замість мікросхеми FTDI використано

мікроконтролер ATmega16U2 (ATmega8U2 в версіях плати R1 і R2). На платі Mega 2560 версії R2 доданий резистор, що підтягує до землі лінію HWB мікроконтролера 8U2. Подібний підхід дає змогу спростити процес оновлення прошивки і перехід пристрою в режим DFU. На платі R3 додані виводи SDA і SCL (біля виведення AREF), а також два нових порта виводу, розташованих біля виводу RESET. Перший - IOREF - дає змогу платам розширення підлаштовуватися під робочу напругу Ардуіно. Даний вивід передбачений для сумісності плат розширення як з 5В Ардуіно на базі мікроконтролерів AVR, так і з 3.3В платами Arduino Due. Другий вивід ні до чого не приєднаний і зарезервований для інших цілей. Також покращена стійкість електричного кола скидання й мікроконтролер ATmega16U2 замінений на 8U2.

Arduino Mega2560 спроектований таким чином, щоб бути сумісним з більшістю плат розширення для Ардуіно Uno, Diecimila і Duemilanove. Для цього цифрові виводи 0 - 13 (а також суміжні з ними виводи AREF і GND), аналогові входи 0 - 5, роз'єм живлення і ICSP-роз'єм на всіх платах розташовані однаково. Крім того, у перерахованих пристроях лінії основного приймача UART з'єднані з одними і тими ж виводами (0 і 1), як і лінії зовнішніх переривань 0 і 1 (виводи 2 і 3 відповідно). Лінії інтерфейсу SPI виведені на роз'єм ICSP на обох платах - як на Mega2560, так і на Duemilanove / Diecimila.

Технічні характеристики плати Arduino Mega 2560:

- мікроконтролер: ATmega2560;
- робоча напруга: 5 В;
- вхідна напруга: (7...20) В;
- цифрові вхідні порти: 54 (з яких 15 забезпечують ШІМ-вихід);
- аналогові вхідні порти: 16;
- потужність постійного струму на вході / виході: 40 мА;
- сила постійного струму для контактів 3.3 В: 50 мА;
- флеш-пам'ять: 256 кБ, з яких 8 кБ використовує завантажувач;
- SRAM: 8 кБ;
- EEPROM: 4 кБ;

- тактова частота 16 МГц;
- довжина: 101 мм;
- ширина: 53,4 мм;

1.3 Огляд периферійних пристроїв

Периферійний пристрій - це допоміжний пристрій, який використовують для надсилання інформації та отримання інформації з мікроконтролера.

Існують три категорії периферійних пристроїв на основі їх взаємозв'язку з мікроконтролером:

1) пристрій вводу, що передає на комп'ютер дані через джойстик, кнопку, клавіатуру, датчик відстані, датчик присутності, датчик температури, датчик вологості, датчик світла, мікрофон, цифрова камера, веб-камера, датчик води, датчик пульсу тощо;

2) пристрій виводу - забезпечує вихід з мікроконтролера, на такі пристрої як світлодіод, монітор, сервопривід, двигун, гучномовець тощо;

3) пристрій вводу/виводу - виконує функції вводу та виводу, через пристрій зберігання даних (дисковий пристрій, USB флеш-диск, карта пам'яті та стрічковий пристрій) та сенсорний екран.

Для реалізації системи контролюваного нагрівання води потрібні такі периферійні пристрої, як: потенціометр, фоторезистор, датчик температури DS18B20, датчик тиску MPXV5050, інфрачервоний модуль, твердотільне реле, драйвер мотору, сервопривід, водяна помпа, RGB – світлодіод, модуль звуку, дисплей LCD 2004, модуль Bluetooth.

1.3.1. Пристрої вводу інформації

1.3.1.1. Потенціометр

Потенціометр - це електричний пристрій, значення опору якого можна змінювати в певних межах [11]. Таким чином можемо регулювати параметри електричних кіл, гнучко налаштовувати їх задля певних умови наприклад, регулювати чутливість датчика або гучність звуку в гучномовці. Потенціометри

набули широкого поширення в схемах регулювання гучності, напруги, контрастності тощо за простоту і зручність.



Рисунок 1.3 - Потенціометр

Залежно від конструкції потенціометри поділяють на два класи: цифрові і аналогові. Основним елементом цифрового потенціометра є резистивні сходи, де на кожному кроці є електронні перемикачі. У конкретний момент часу відбувається комутація тільки одного електронного вимикача, що задає певне значення опору. Внаслідок деякої кількості кроків визначено діапазон регулювання потенціометра. Аналоговий потенціометр може змінювати значення опору безперервно, але, як правило, у більш вузькому діапазоні і резистор має мати великі габарити.

У переважній більшості ситуацій в проектах Ардуіно використовують цифрові потенціометри. Найчастіше вони є інтегральними схемами з положенням цифрового показника по центру шкали.

Підключення потенціометра до Ардуіно показано на рис.1.4.

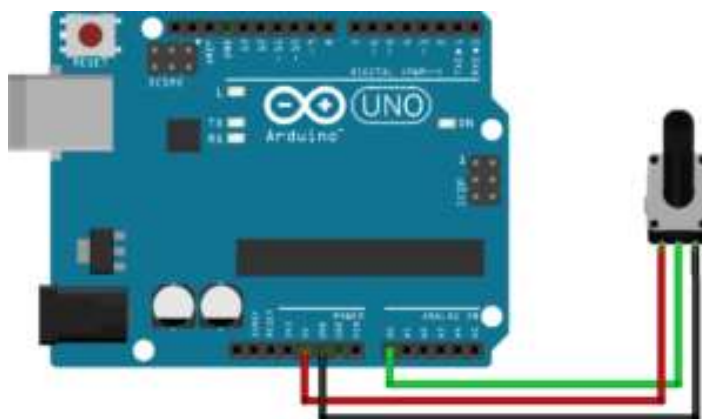


Рисунок 1.4 – Підключення потенціометра

Для цього три виводи потенціометра необхідно з'єднати з зазначеними виводами плати:

- чорний - GND;
- червоний - живлення 5В;
- зелений - від центрального виведення до аналогового входу А0.

За змінення положення вала потенціометра, відбувається зміна значення опору, яке спричинює зміну показника на нульовому вході плати Ардуіно. Зчитування отриманого значення напруги аналогового імпульсу відбувається в скетчі за команди `analogRead ()`.

У плату Ардуіно вбудовано аналого-цифровий перетворювач, здатний зчитувати напругу і переводити його в цифрові показники зі значенням від 0 до 1023. В разі повороту валу до кінцевого значення в одному з двох можливих напрямків, напруга на вході дорівнює нулю, і, отже, генерована напруга дорівнює 0 В. За повороту валу до кінця у протилежному напрямку на вхід надходить напруга значенням 5В, і числове значення становитиме 1023.

1.3.1.2. Фоторезистор

Фоторезистор - це напівпровідниковий прилад, опір якого змінюється залежно від стану освітлення чутливої поверхні [12]. Конструктивне реалізація може бути різною. Поширені елементи показано на рисунку 1.5. При цьому для роботи у специфічних умовах застосовують фоторезистори в металевому корпусі з віконцем, через яке потрапляє світло на чутливу поверхню. Умовне графічне позначення наведено на рисунку 1.6

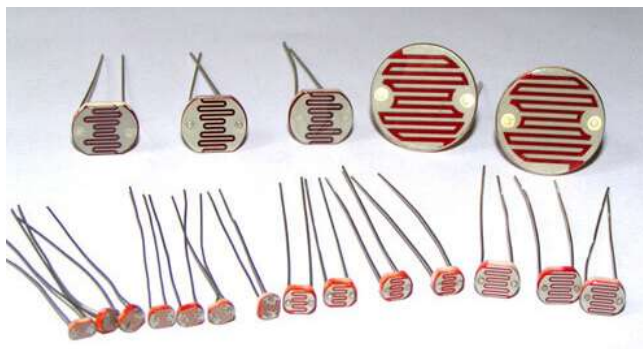


Рисунок 1.5 - Фоторезистори

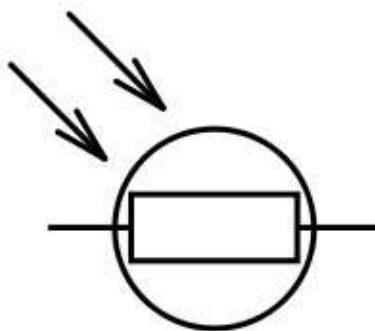


Рисунок 1.6 – Графічне позначення фоторезистору

Зміна опору під впливом світлового потоку називають фоторезистивним ефектом. Принцип дії фоторезистора такий: між двома провідними електродами розташовано напівпровідник. Якщо напівпровідник не освітлений - його опір великий. Коли ця область освітлена її провідність різко зростає, а опір відповідно падає.

Матеріали напівпровідника можуть використовувати такі як: сульфід кадмію, сульфид свинцю, сполуки кадмію тощо. Від вибору матеріалу фоторезистора залежить його спектральна характеристика, а саме діпазон кольорів (довжин хвиль) під час освітлення якими буде коректно змінюватися опір елемента. Тому під час вибору фоторезистора, потрібно враховувати в якому спектрі він працює. Наприклад, для ультрафіолетових чутливих елементів потрібно підбирати види випромінювачів, спектральні характеристики яких прийнятні для фоторезистора. Спектральні характеристики різних матеріалів наведено на рисунку 1.7.

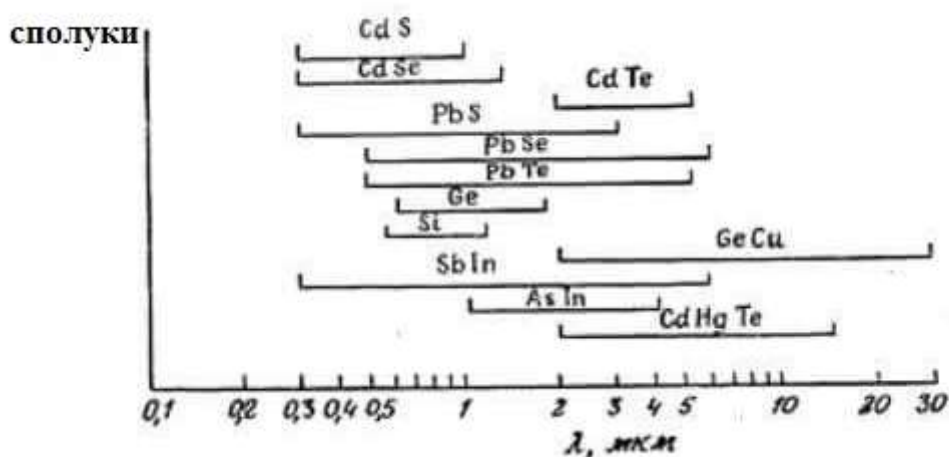


Рисунок 1.7 – Спектральні характеристики матеріалів

У фоторезисторів немає р-п переходу, тому не має значення, в якому напрямку протікає струм. Перевірити фоторезистор можна мультиметром в режимах вимірювання опору у разі освітленого та затемненого елемента.

Залежність опору від освітленості наведено на рисунку 1.8:

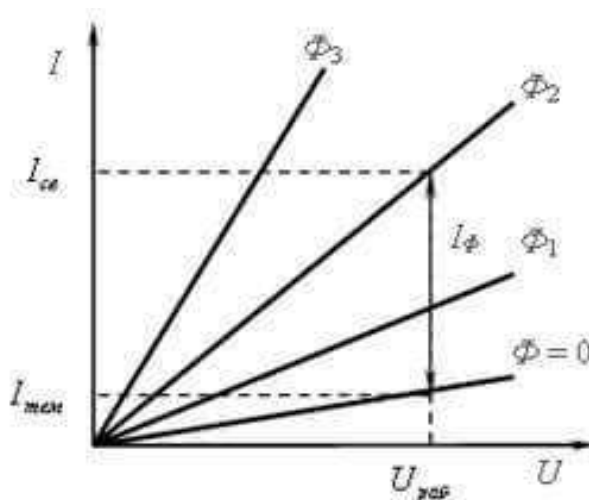


Рисунок 1.8 – Залежність опору від освітленості

Струм змінюється за певному значенні напруги в залежності від кількості світла, де $\Phi = 0$ - темрява, а Φ_3 - яскраве світло. Змінення сили струму від значення постійній напрузі наведено на рисунку 1.9:

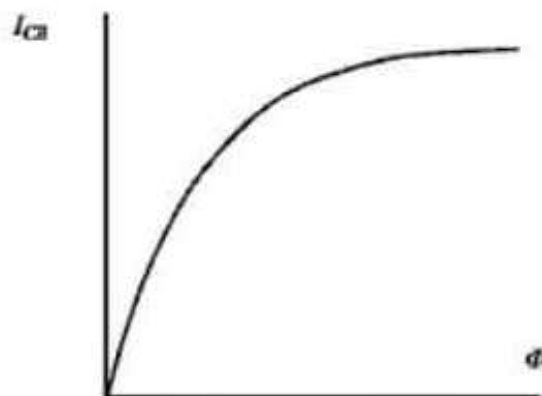


Рисунок 1.9 – Зміна струму за постійній напрузі

Залежність опору фоторезистора від освітленості наведено на рисунку 1.10:

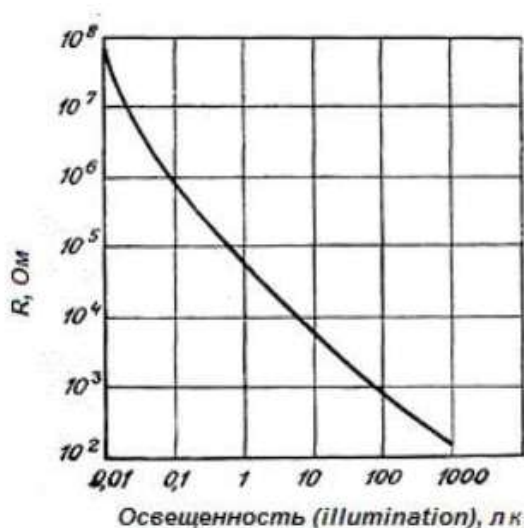


Рисунок 1.10 - Залежність опору від освітленості

Характеристики фоторезисторів:

- темновий опір - це опір фоторезистора під час відсутності світлового потоку;
- інтегральна фоточутливість - реакція елемента на зміну сили струму через нього на зміну світлового потоку: $S = I_{\phi} / \Phi$, де I_{ϕ} - фотострум, а Φ - світловий потік.

Фоторезистори мають певну інерційність, тобто його опір змінюється з деякою невеликою затримкою. Цей параметр характеризує гранична частота (частота синусоїдального сигналу, що модулює світловий потік елемент, за якій

чутливість елемента знижується в 1,41) Швидкодія компонентів зазвичай є в межах десятків мікросекунд 10^{-5} с, тому їх застосування часто не є доцільним.

Схема підключення фоторезистора до Ардуіно наведена на рисунку 1.11.

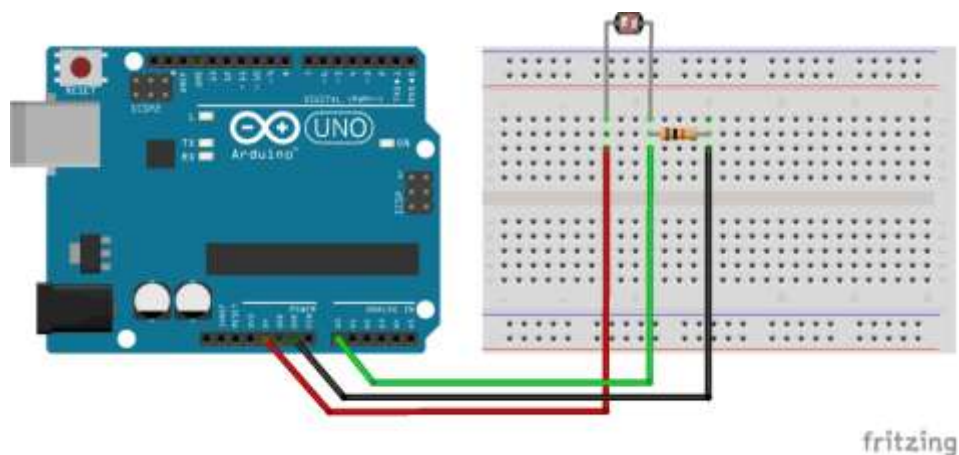


Рисунок 1.11 – Схема підключення фоторезистора

1.3.1.3. Датчик температури DS18B20

DS18B20 - це цифровий температурний датчик, що має безліч корисних функцій. DS18B20 - це мікроконтролер, який може зберігати значення вимірювань, сигналізувати про вихід температури за встановлені межі, міняти точність вимірювань, спосіб взаємодії з контролером, тощо [13]. Все це в дуже невеликому корпусі, який, до того ж, доступний в водонепроникному виконанні.

PIN ASSIGNMENT

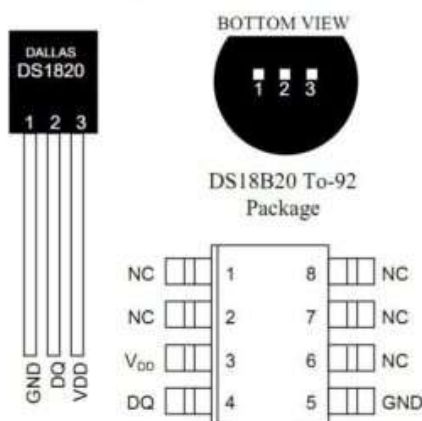


Рисунок 1.12 - Датчик температури DS18B20 [13]

Мікросхема має три виходи, з яких для даних використано тільки один, а два інших - проводи електроживлення. Число проводів можна скоротити до двох,

якщо використовувати, так звану схему «паразитного електроживлення» і з'єднати Vdd з землею. До одного проводу з даними можна підключити відразу декілька датчиків DS18B20 і в платі Ардуїно застосовано лише один цифровий вхід.

Види корпусів DS18B20

Температурний датчик DS18B20 має різноманітні види корпусу. Можна вибрати один з трьох - 8-Pin SO (150 mils), 8-Pin μ SOP, і 3-Pin TO-92. Останній є найбільш поширеним, його виготовляють у спеціальному вологостійкому корпусі, так що його можна використовувати під водою. У кожного датчика є 3 контакти. Для корпусу TO-92 потрібно враховувати колір проводів: чорний - земля, червоний - електроживлення і білий/жовтий/синій - сигнал.

Особливості цифрового датчика DS18B20:

- похибка вимірювання не більше $0,5^{\circ}\text{C}$ (для температур від -10°C до $+85^{\circ}\text{C}$), що забезпечує достатньо точно визначення значення температури; додаткове калібрування не є потрібно;
- температурний діапазон вимірювань є в межах від -55°C до $+125^{\circ}\text{C}$;
- датчик живлять напругою від 3,3В до 5В;
- можна програмно задати максимальну роздільну здатність до $0,0625^{\circ}\text{C}$, найбільший дозвіл 12 біт;
- присутня функція тривожного сигналу;
- кожен пристрій має свій унікальний серійний код;
- не потрібні додаткові зовнішні елементи;
- можна підключити відразу до 127 датчиків до однієї лінії зв'язку.
- інформація передається по протоколу 1-Wire.
- для приєднання до мікроконтролеру потрібні лише 3 дроти;
- є можливим режим «паразитного живлення» (рис 1.13) - (це живлення організовано від лінії зв'язку; для підключення потрібні 2 дроти; але в цьому режимі не гарантовано коректну роботу за температур вище 100°C). Цей режим зручно застосовувати для додатків з віддаленим температурним датчиком.

Пам'ять датчика складено з двох видів: оперативної і незалежній - SRAM і EEPROM. В останню записують регістри конфігурації і регістри TH, TL, які можуть використовувати як регістри загального призначення, якщо не використано для визначення діапазону допустимих значень температури.

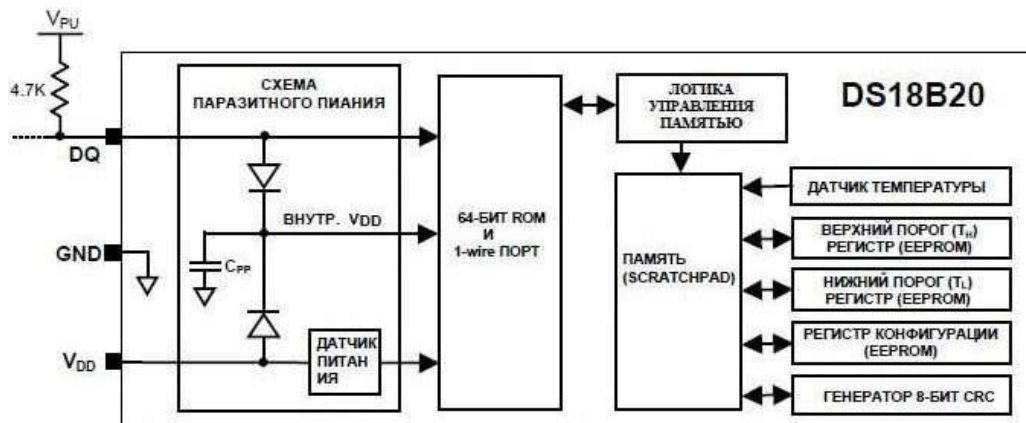


Рисунок 1.13 – Схема підключення «паразитного живлення»

Основним завданням DS18B20 є визначення температури і перетворення отриманого результату у цифрову форми. Можна задати необхідну роздільну здатність, за кількості біт точності: 9, 10, 11 і 12. У цих ситуаціях відповідно маємо $0,5^{\circ}\text{C}$, $0,25^{\circ}\text{C}$, $0,125^{\circ}\text{C}$ і $0,0625^{\circ}\text{C}$.

Під час включення живлення датчик є в стані спокою. Для початку вимірювання контролер Ардуіно виконує команду «перетворення температури». Отриманий результат збережено в 2 байтах регістра температури, після чого датчик повернено в первинний стан спокою. Якщо схема підключена у режимі зовнішнього живлення, мікроконтролер регулює стан конвертації. Під час виконання команди лінія є в низькому стані, після закінчення програми лінія переходить в високий стан. Такий метод не допустимо під час живленні від «паразитної ємності», бо на шині постійно має зберігатися високий рівень сигналу.

Отримані температурні вимірювання зберігаються в SRAM датчика. 1 і 2 байти зберігають отримане значення температури, 3 і 4 зберігають межі вимірювання, 5 і 6 зарезервовані, 7 і 8 використано для високоточного визначення температури, останній 9 байт зберігає стійкий до завад CRC код.

Підключення DS18B20 до Arduino

DS18B20 є цифровим датчиком. Цифрові датчики передають значення вимірюваної температури у вигляді певного двійкового коду, який надходить на цифровий або аналоговий вхід Ардуїно з наступним декодуванням. Коди можуть бути різними, ds18b20 працює по протоколу даних 1-Wire.

Обмін інформацією в 1-Wire відбувається за такими операціями:

- ініціалізація - визначення послідовності сигналів, з яких починається вимір і інші операції. Провідний пристрій подає імпульс скидання, після цього датчик подає імпульс присутності, повідомляє про готовність до виконання операції;
- запис даних - відбувається передача байта даних в датчик;
- читання даних - відбувається прийом байта з датчика.

Схема підключення цифрового датчика DS18B20 наведена на рисунку 1.14.

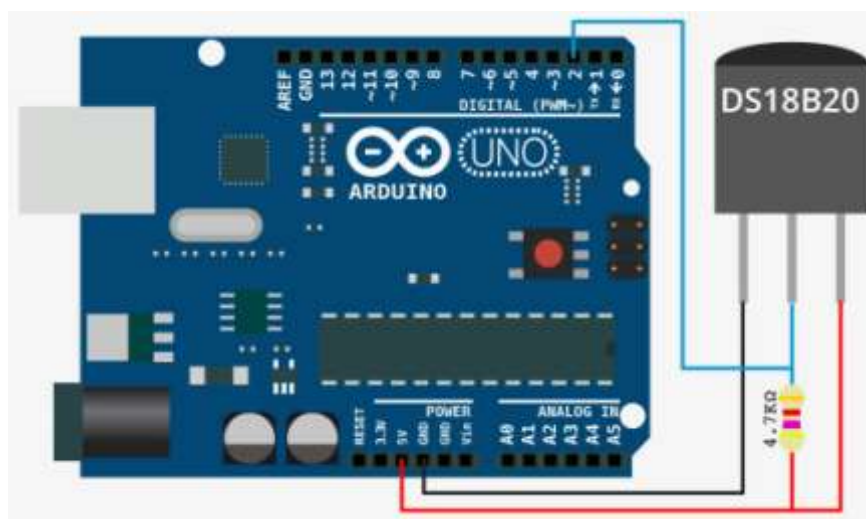


Рисунок 1.14 - Схема підключення цифрового датчика DS18B20

- GND з термодатчика приєднується до GND Ардуїно;
- vdd підключається до 5V;
- data - до будь-якого цифрового піну.

У режимі паразитного живлення контакт Vdd з датчика підключається до GND на Ардуїно. Роботу в паразитному режимі краще не використовувати без необхідності, так як можуть погіршитися швидкодія і стабільність.

Алгоритм отримання інформації про температуру в скетчі складається з етапів:

- визначення адреси датчика, перевірка його підключення;

- на датчик подано команда з вимогою прочитати температуру і викласти виміряне значення в регістр. Процедура відбувається довше за інших, на неї необхідно приблизно 750 мс;
- подана команда на читання інформації з регістра і відправка отриманого значення в «монітор порту»;
- якщо потрібно, то виконують конвертацію: °C / °F.

1.3.1.3. Датчик тиску MPXV5050

Датчик тиску MPXV5050 розрахований на 50 кПа. Це еквівалентно водяному стовпу висотою 5 метрів.

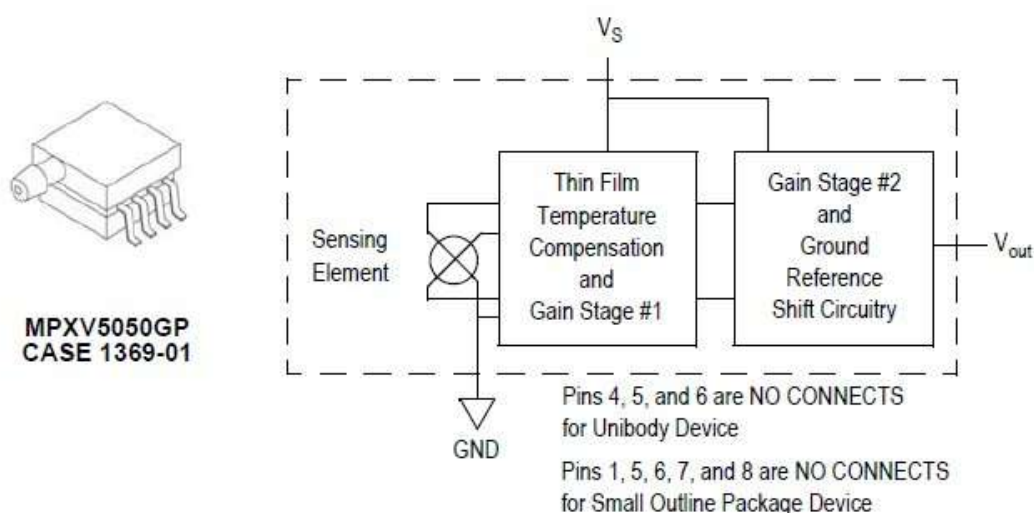


Рисунок 1.15 - Датчик тиску MPXV5050 [14]

Технічні характеристики наведено у таблиці 1.1.

Таблиця 1.1 - Технічні характеристики датчика тиску MPXV5050

Тип датчику	Інтегрований
Тип тиску	Відносний
P_{OP} , кПа	від 0 до 50
P_{MAX} , кПа	200
V_{FSS} (тип.), мВ	4.5

Нелінійність, %	2.5
Чутливість, мВ/кПа	90
$V_{CC}, \text{В}$	від 4.75 до 5.25
$I_{CC}, \text{мА}$	7
$T_A, ^\circ\text{C}$	від -40 до 125

Конструкція вимірювання тиску [14]

У воду опускають мідну трубку маленького діаметру майже до самого дна. Мідь обрана, тому що вона має високу електро- та теплопровідність, корозійну стійкість і відмінно сприймає гарячу і холодну обробку тиском. Вона стійка до атмосферної корозії. Чиста прісна вода майже не діє на мідь [15]. За підключення нижній зріз мідній трубки відкрито, а до верхнього підключають датчик тиску. У результаті датчик фіксує тиск водяного стовпа на нижньому зрізі труби (1 кПа - 10 см). Дійсно, за стисненням повітря вода дещо заходить всередину труби. Тому датчик фіксує тиск не за нижньому зрізу труби, а на рівні H_1 (рис.1). Якщо відомо на скільки зростає тиск всередині труби у порівнянні з атмосферним, можна доволі точно обчислити висоту цієї різниці. Для цього потрібно, щоб труба була постійного перетину.

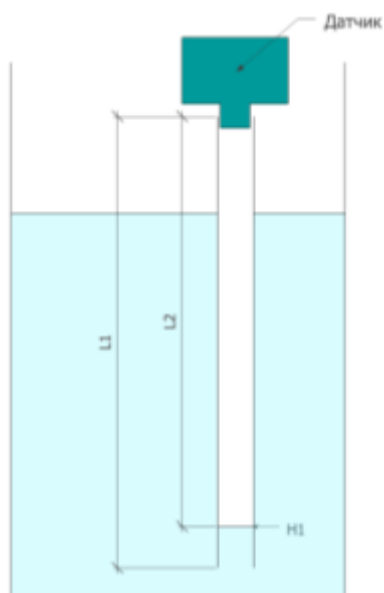


Рисунок 1.16 – Конструкція вимірювання тиску

P - виміряний тиск всередині трубки, P_a - початковий атмосферний тиск,

V_1 - початковий об'єм повітря, V_2 - об'єм після занурення,

L_1 - довжина всієї трубки, L_2 - довжина трубки від датчика до H_1

Оскільки кількість повітря в трубці не змінюється, то

$$P_a * V_1 = P * V_2 \text{ [16]},$$

Оскільки перетин (площа) труби постійна, то

$$P_a * L_1 = P * L_2;$$

$$L_2 = (P_a * L) / P.$$

Тиск водяного стовпа висотою 1 метр дорівнює 10 кПа, а атмосферний тиск орієнтовно 100 кПа [17], тому можна порахувати на скільки стиснено повітря, якщо опустити трубку завдовжки 1 метр, на глибину 1 метр.

$$L_2 = (100 * 1\text{м}) / (100 + 10) = 0.9 \text{ м}$$

Тобто вода буде на рівні 10 см від нижньому зрізу труби.

1.3.1.5. Інфрачервоний модуль

ІЧ приймач і інфрачервоний пульт дистанційного керування - найпоширеніший і простий спосіб управління електронною апаратурою. Інфрачервоний спектр випромінювання непомітний людським оком, але він відмінно приймається ІК приймачами, які вбудовані в електронні прилади. Модулі Arduino IRremote створено для управління різною технікою в прямої видимості [18].



Рисунок 1.17 - ІЧ приймач і інфрачервоний пульт

Широке застосування ІЧ випромінювачів стало можливим через їх низьку вартість, простоту і зручність у використанні. ІЧ випромінювання лежить в діапазоні від 750 до 1000 мкм - це найближча частина спектра до видимого світла. В області інфрачервоного випромінювання можуть змінюватися оптичні властивості різних матеріалів. Деякі скла, наприклад, стають непрозорими для інфрачервоних променів, парафін ж навпаки прозорий в ІЧ спектрі.

Реєструється випромінювання за допомогою спеціальних фотоматеріалів, на основі яких виготовляються приймачі. Джерелом інфрачервоного випромінювання крім нагрітих тіл (Сонця, ламп розжарювання або свічок), можуть бути твердотільні прилади – ІЧ світлодіоди, лазери. Випромінювання в інфрачервоному діапазоні має низку особливостей, завдяки яким їх зручно використовувати в пультах:

- твердотільні випромінювачі (ІЧ світлодіоди) коштують дешево і вони компактні;
- інфрачервоні промені не сприймаються і не фіксуються людським оком;
- ІЧ приймачі також дешево коштують, і вони мають невеликі розміри;
- малі перешкоди, так як передавач і приймач налаштовані на одну частоту;
- відсутній негативний вплив на здоров'я людини;
- високий показник відбиття від більшості матеріалів;

- ІЧ випромінювачі не впливають на роботу інших пристроїв;

Робота пульта здійснюється наступним чином. При натисканні кнопки відбувається кодування сигналу в інфрачервоному світлі, приймач приймає його і виконує потрібні дії. Інформація кодується у вигляді логічної послідовності пакетів імпульсів з певною частотою.

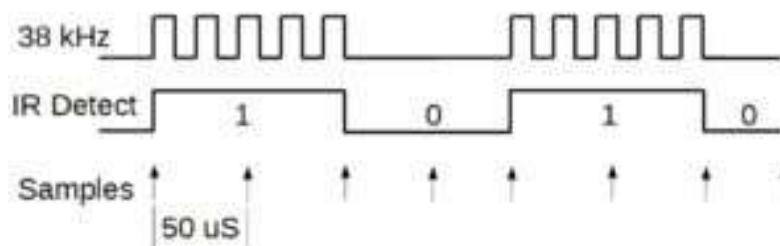


Рисунок 1.18 - Послідовність пакетів імпульсів

Приймач отримує цю послідовність і виконує демоделювання даних. Для прийому сигналу використовується мікросхема, в якій містяться фотоприймач (фотодіод), підсилювачі, смуговий фільтр, демодулятор і вихідний транзистор. Також в ній встановлені фільтри - електричний і оптичний. Працюють такі пристрої на відстані до 40 метрів. Існує величезна безліч різних приймачів, але краще використовувати TSOP312 або інші відповідні для Ардуїно. Дані від пульта до приймача можуть передаватися по протоколу RC5 або NEC. Центральний контакт - це земля, зліва - вихід на мікроконтролер, праворуч - живлення.

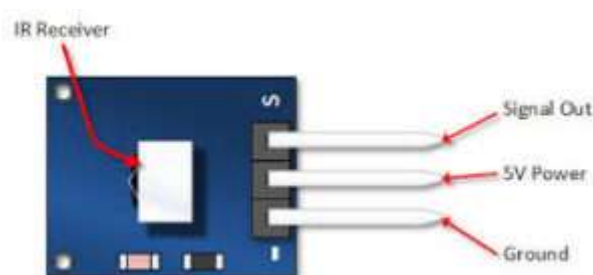


Рисунок 1.19 – Виводи ІЧ приймача

Для зручності можна використовувати готові модулі ІЧ приймача.



Рисунок 1.20 - Модуль ІЧ приймача.

Схема підключення датчика до Ардуїно зображена на рисунку 1.21:



Рисунок 1.21 – Схема підключення до Ардуїно

1.3.2. Пристрої виводу інформації

1.3.2.1. Реле для комутації

Реле дозволяє підключити пристрої, що працюють в режимах з відносно великими струмами або напруги. Не можна безпосередньо підключити до плати Arduino потужні насоси, двигуни або звичайну лампу розжарення, тому що плата не призначена для такого навантаження.

Реле - це шлюз, який дозволяє з'єднати разом електричні кола з абсолютно різними параметрами.

Реле характеризується наступними параметрами:

- напруга або струм спрацьовування;
- напруга або струм відпускання;

- час спрацювання і відпускання;
- робочі струм і напруга;
- внутрішній опір.

Залежно від типу внутрішніх механізмів і особливості пристрою можна виділити дві основні групи реле: електромеханічні реле (включення за допомогою електромагніту) і твердотільні реле (включення через спеціальні напівпровідникові компоненти).

Електромагнітне реле

Електромагнітне реле - це електричний пристрій, який механічним шляхом замикає або розмикає коло навантаження за допомогою магніту. Він складається з електромагніту, рухомого якоря і перемикача [19]. Електромагніт - це провід, який намотаний на котушку з феромагнетика. У ролі якоря виступає пластина з магнітного матеріалу. В деякі моделі пристрою можуть бути вбудовані додаткові електронні компоненти: резистор для більш точного спрацювання реле, конденсатор для зменшення перешкод, діод для усунення перенапруги.



Рисунок 1.22 - Електромагнітне реле

Працює реле внаслідок електромагнітної сили, що виникає в сердечнику при подачі струму по виткам котушки. У початковому стані пружина утримує якір. Коли подається керуючий сигнал, магніт починає притягувати якір і замикає або розмикає коло. При відключенні напруги якір повертається в початкове положення. Джерелами керуючого напруги можуть бути датчики (тиску,

температури і інші), електричні мікросхеми та інші пристрої, які подають малий струм або малу напругу.

Електромагнітне реле застосовується в схемах автоматики, при управлінні різними технологічними установками, електроприводами і іншими пристроями. Реле призначене для регулювання напруги і струмів, може використовуватися як накопичувач або перетворюючий пристрій, також може фіксувати відхилення параметрів від нормальних значень.

Класифікація електромагнітних реле:

- керуючий струм може бути як постійним, так і змінним. У першому випадку пристрій може бути нейтральним або поляризованим. Для змінного струму якір виконується з електротехнічної сталі, щоб зменшити втрати;
- якірне або герконовое реле. Для якірного процес замикання і розмикання відбувається за допомогою переміщення якоря, для герконового характерна відсутність сердечника, магнітне поле впливає на електрод з контактами;
- швидкодія - до 50 мс, до 150 мс і від 1 с;
- захисне покриття - герметизовано, закрите і відкрите.

У порівнянні з напівпровідниковими пристроями електромагнітне реле має переваги - воно коштує недорого, можливість комутації великого навантаження при невеликому розмірі пристрою, мале виділення тепла на котушці. З недоліків можна виділити повільне спрацьовування, перешкоди і складність комутації індуктивних навантажень.

Твердотільні реле

Твердотільні реле - це модульний напівпровідниковий пристрій, що виробляється по гібридній технології [20] н-мост. Він вважається альтернативою електромагнітним реле. У складі реле є транзистори, сімістори або тиристори.



Рисунок 1.23 - Твердотільне реле

У порівнянні з електромагнітними пристроями твердотільні реле мають ряд переваг:

- довгий термін експлуатації;
- швидкодія;
- малі розміри;
- відсутні сторонні шуми, акустичні перешкоди, брязкіт контактів;
- низьке споживання енергії;
- якісна ізоляція;
- стійкість до вібрації і ударів;
- немає дугового розряду, що дозволяє працювати у вибухонебезпечних місцях.

Принцип дії реле такий: подається керуючий сигнал на світлодіод, відбувається гальванічна розв'язка керуючих та комутованих кіл, потім сигнал переходить на фотодіодну матрицю. Напруга регулює силовим ключем.

Твердотільні реле також мають кілька недоліків. По-перше, при комутації відбувається нагрів пристрою. Підвищення температури пристрою призводить до обмеження регульованого струму - при температурах, що перевищують 60 градусів, зменшується величина струму, максимальна робоча температура 80 градусів.

Твердотільні реле класифікуються за такими ознаками:

- тип навантаження - однофазні та трифазні;

- спосіб управління - комутація відбувається постійною чи змінною напругою, або ручного управління;
- метод комутації: контроль переходу через нуль (застосовується для слабоіндуктивних, ємнісних і резистивних навантажень), довільне включення (індуктивні і резистивні навантаження, яким необхідно миттєве спрацювання) і фазове управління (зміна вихідної напруги, регулювання потужності, управління лампами розжарювання).

Найбільш поширене реле для плати Ардуіно виконано у вигляді модуля, наприклад, SONGLE SRD-05VDC. Пристрій управляється напругою 5 В, може комутувати до 10 А 30 В DC та 10 А 250 В AC.

Реле - це дві не пов'язані між собою кола - керуючого кола A1 і A2 і керована 1, 2 і 3. Схема реалізації реле зображена на рисунку 1.24.

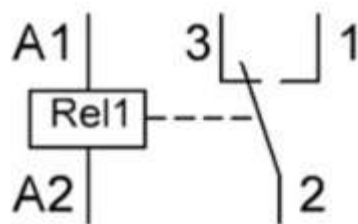


Рисунок 1.24 - Схема реалізації реле

Між A1 і A2 є металевий сердечник. Якщо пустити по ньому електричний струм, до нього притягнеться якір (2). 1, 3 - нерухомі контакти. За відсутності струму якір є близько контакту 3.

Схема підключення реле до Ардуіно зображена на рисунку 1.25

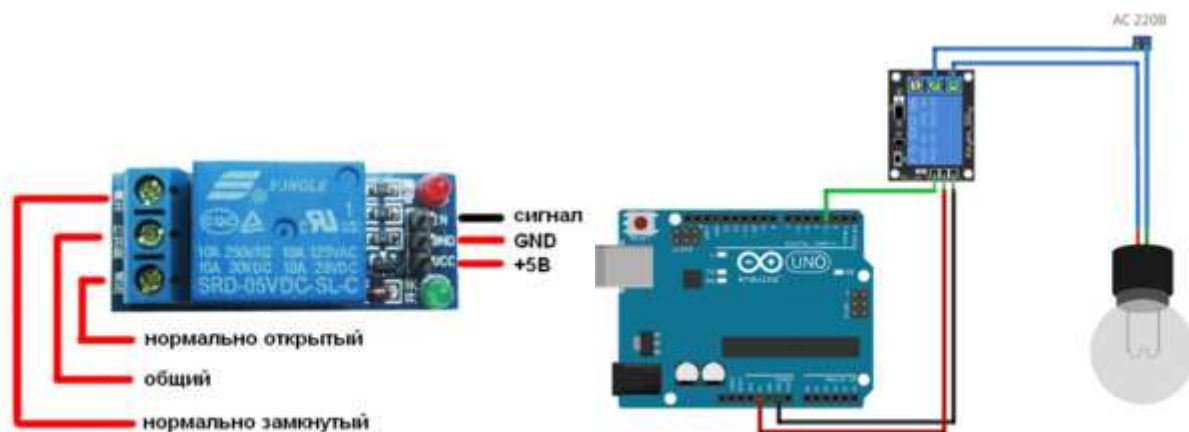


Рисунок 1.25 – Схема підключення реле до Ардуіно

Реле має всього 3 контакти, підключаються вони до Ардуїно наступним чином: GND до GND, VCC до + 5V, In до цифрового входу. Вхід реле - інвертований, так що високий рівень на In вимикає котушку, а низький – вмикає.

Світлодіоди потрібні для індикації. Під час вмикання червоного LED1 подається напруга на реле, при вмиканні зеленого LED2 відбувається замикання. Коли включається мікроконтролер, транзистор у закритому стані. Для його відкриття на базу потрібен мінус, який подається за допомогою функції `digitalWrite (pin, LOW)`; Транзистор стає у відкритий стан, протікає струм через коло, реле спрацьовує. Щоб його вимкнути, на базу подається плюс за допомогою `digitalWrite (pin, HIGH)`.

1.3.2.2. Драйвер двигуна

Мікроконтролер Ардуїно має обмеження по силі струму приєднаних до нього навантажень. Для усього мікроконтролера це 800 мА, а для кожного окремого виводу – 40 мА. Неможливо підключити безпосередньо до Arduino Uno, Mega або Nano двигун постійного струму, сервопривід або водяну помпу. Будь-який з цих периферійних пристроїв в момент запуску або зупинки створить пікові підйоми струму, що перевищують цю межу.

Для підключення цих периферійних пристроїв треба використовувати драйвери двигуна, які побудовані на спеціальній схемі з Н-мостом [21]. Існують як мікросхеми, що містять два або більше Н-моста, так і окремі модулі та плати розширення, побудовані на цих мікросхемах.

Мікросхема або плата розширення Motor Shield

Motor Shield - плата розширення для Ардуїно, яка забезпечує роботу двигунів постійного струму і крокових двигунів. Найпопулярнішими платами Motor Shield є схеми на базі чіпів L298N і L293D, які можуть керувати кількома двигунами. На платі встановлений комплект наскрізних колодок Ардуїно Rev3, що дозволяють встановлювати інші плати розширення. Також на платі є можливість вибору джерела напруги. Motor Shield може житися як від Ардуїно,

так і від зовнішнього джерела. На платі є світлодіод, який показує, чи працює пристрій. Використання драйвера є дуже простим і надійним.

Принцип дії Н-моста

Принцип роботи драйвера двигуна заснований на принципі роботи Н-моста. Н-міст є електронною схемою, яка складається з чотирьох ключів з навантаженням. Назва мосту з'явилася тому що нагадує букву Н конфігурації схеми. Схема моста зображена на рисунку 1.26.

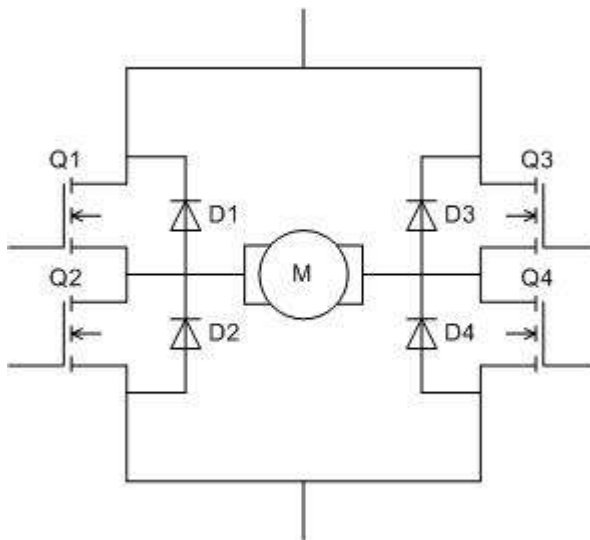


Рисунок 1.26 – Електронна схема Н-мосту

Q1, Q2, Q3, Q4 - польові, біполярні або IGBT транзистори. Останні використовуються в високовольтних мережах. Біполярні транзистори практично не використовуються, вони можуть бути присутніми в малопотужних схемах. Для великих струмів беруть польові транзистори з ізолюваним затвором. Ключі не повинні бути замкнуті разом одночасно, щоб не сталося короткого замикання джерела. Діоди D1, D2, D3, D4 обмежувальні, зазвичай використовуються діоди Шоттки.

За допомогою зміни стану ключів на Н-мосту можна регулювати напрямок руху і гальмувати мотори. У таблиці наведені основні стани та відповідні їм комбінації на виводах.

Q1	Q2	Q3	Q4	Стан
1	0	0	1	Поворот мотора праворуч
0	1	1	0	Поворот мотора ліворуч
0	0	0	0	Вільне обертання
0	1	0	1	Гальмування
1	0	1	0	Гальмування
1	1	0	0	Коротке замикання
0	0	1	1	Коротке замикання

Драйвер двигуна L298N

Модуль використовується для управління кроковими двигунами з напругою від 5 до 35 В. За допомогою однієї плати L298N [22] можна керувати відразу двома двигунами. Найбільше навантаження, яку забезпечує мікросхема, досягає 2 А на кожен двигун. Якщо підключити двигуни паралельно, це значення можна збільшити до 4 А.



Рисунок 1.27 – Драйвер двигуна L298N

Виводи мікросхеми L298N зображено на рисунку 1.28:



Рисунок 1.27 – Виводи мікросхеми L298N

- Vcc - використовується для підключення зовнішнього живлення;
- 5V;
- земля GND;
- IN1, IN2, IN3, IN4 - використовується для плавного регулювання швидкості обертання двигуна;
- OUT1, OUT2 - використовується для виходу з першого двигуна;
- OUT3, OUT4 - використовується для виходу з другого двигуна;
- S1 - перемикає живлення схеми: від зовнішнього джерела або від внутрішнього перетворювача;
- ENABLE A, B - потрібні для роздільного управління каналами. Використовуються в двох режимах - активний, при якому каналами управляє мікроконтролер і є можливість зміни швидкості обертання, і пасивний, в якому неможливо управляти швидкістю двигунів (встановлене максимальне значення).

При підключенні двох двигунів, потрібно перевірити, щоб у них була однакова полярність. Якщо полярність різна, то при завданні напрямку руху вони будуть обертатися в протилежні сторони.

Підключення L298N до Arduino зображено на рисунку 1.28

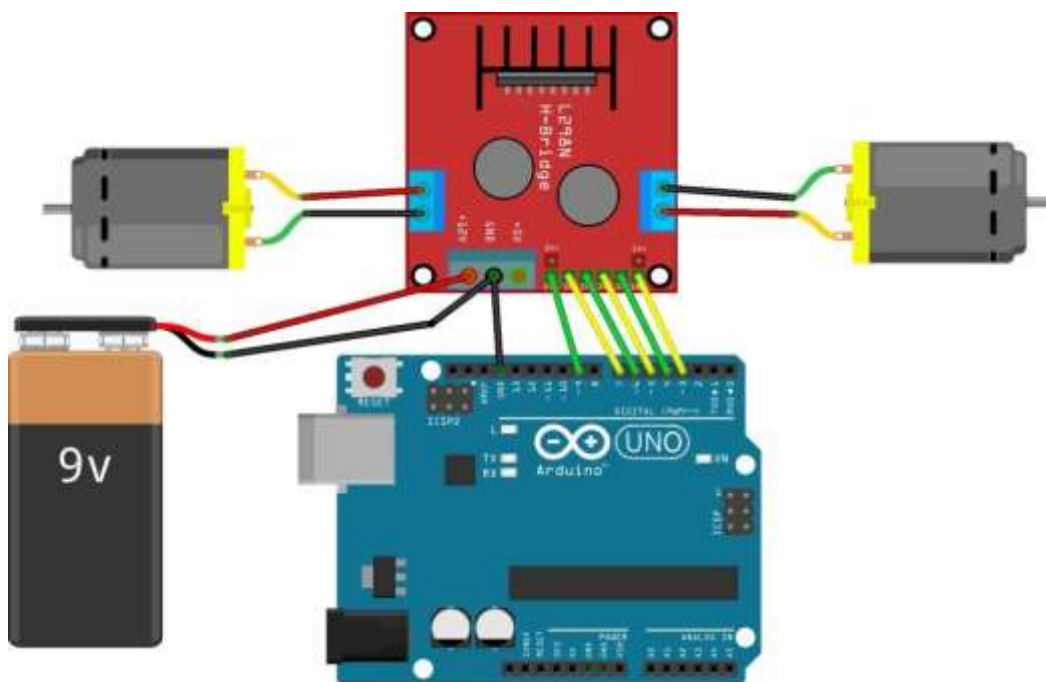


Рисунок 1.28 – Підключення L298N до Arduino

Напрямок обертання задається сигналами HIGH і LOW на кожен канал. Щоб керувати швидкістю, використовуються ШІМ-сигнали.

1.3.2.3. Сервопривід

Сервопривіди (серводвигун) використовують там, де існує потреба в точному русі або у точному положенні валу. Такі датчики використовують не як високошвидкісний додаток, а для малої швидкості, середнього крутного моменту та точного положення. Ці двигуни використовуються в робототехнічних машинах, системах управління польотом та системах управління. Для прикладу серводвигуна обрана модель SG90 [23].



Рисунок 1.29 – Сервопривід SG90

Серводвигуни доступні в різних формах і розмірах. Серводвигун має три дроти: один для позитивної напруги, інший для землі, а останній - для установки положення. Червоний дріт підключається до джерела живлення, чорний дріт до землі, а останній дріт відповідно до керуючого сигналу.

Серводвигун - це комбінація двигуна постійного струму, системи управління положеннями та передач. Положення валу двигуна постійного струму регулюється електронікою керування в сервоприводі, виходячи з співвідношенням податливості сигналу ШІМ до виводу сигналу.

Керуюча електроніка регулює положення валу, керуючи двигуном постійного струму. Ці дані стосовно положення валу надсилаються через контактний сигнал. Дані про положення для контролера повинні надсилатися у формі ШІМ-сигналу через штифт сигналу сервомотора. Частота сигналу ШІМ може змінюватися залежно від типу сервомотора.

Щоб вказати серводвигуну бажане положення, по призначеному для цього проводу необхідно надсилати керуючий сигнал. Керуючий сигнал - імпульси постійної частоти і змінної ширини.

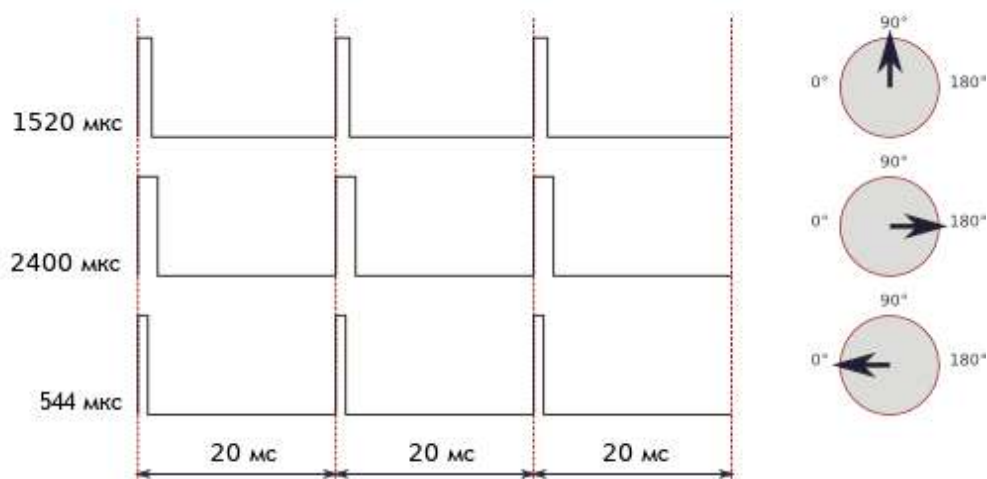


Рисунок 1.30 – Відношення довжини імпульсу до кута повороту

Те, яке положення повинен зайняти сервопривід, залежить від довжини імпульсів. Коли сигнал надходить в керуючу схему, наявний в ній генератор імпульсів виробляє свій імпульс, тривалість якого визначається через потенціометр. Інша частина схеми порівнює тривалість двох імпульсів. Якщо тривалість різна, включається електромотор. Напрямок обертання визначається

тим, який з імпульсів коротше. Якщо довжини імпульсів рівні, електромотор зупиняється.

Найчастіше імпульси виробляються з частотою 50 Гц. Це означає, що імпульс випускається і приймається раз в 20 мс. Зазвичай при цьому тривалість імпульсу в 1520 мкс означає, що сервопривід повинен зайняти середнє положення. Збільшення або зменшення довжини імпульсу змусить сервопривід повернутися за годинниковою або проти годинникової стрілки відповідно. При цьому існують верхня і нижня межа тривалості імпульсу. У бібліотеці Servo для Arduino за замовчуванням виставлені наступні значення довжин імпульсу: 544 мкс - для 0° і 2400 мкс - для 180° .

Характеристики серводвигуна SG-90:

- вага двигуна: 9 грам;
- розміри: 21.5мм x 11.8мм x 22.7мм;
- напруга живлення: 4,8 ... 6 В;
- момент (зусилля): 1.2кг * см при напрузі 4,7 В;
- час повороту на 60° : 0,12сек. при напрузі 4,7 В;
- робоча температура: -30°C ... $+60^\circ\text{C}$;
- довжина кабелю: 23 см;
- матеріал шестерень: нейлон;
- механіка: аналогова.

1.3.2.4. Водяна помпа

Водяна міні помпа - компактний насос для перекачування води в різні ємності. Дозволяє перекачувати рідину з досить високою швидкістю, до 120 літрів на годину. Застосовується для поливу рослин, акваріумах, фонтанах, тощо. Використовуючи цей насос спільно з різними датчиками можна створити систему автоматичного поливу і підтримання життєдіяльності рослин. Може використовуватися в гідропоніці і аквапоніці. В моєму випадку, використано для

підтримки води в пристрої нагрівання води. Завдяки малій напрузі 2.5-6 вольт її можна підключити до сонячної батареї.



Рисунок 1.31 – Водяна помпа

Технічні характеристики:

- напруга живлення: 2,5 - 6 В;
- потужність: 0.4 - 1.5 Вт;
- струм залежить від напруги живлення;
- швидкість перекачування рідини: до 2 л / хв або 120 л / год;
- матеріал корпусу: пластик, загерметизований;
- максимальна висота водяного стовпа: 0.4-1.1 м;
- може перекачувати масло і воду.

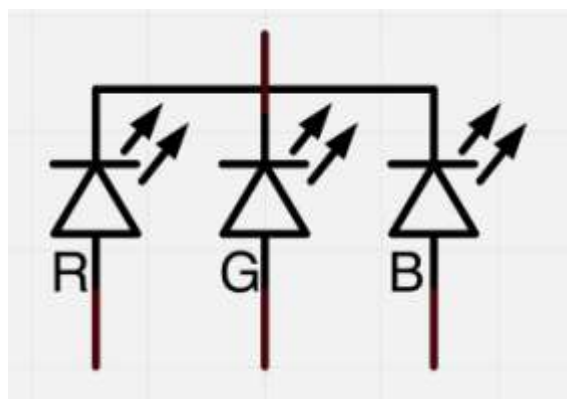
1.3.2.5. RGB – світлодіод

У RGB світлодіоди чотири ноги. По одному позитивному контакти на кожен світлодіод і один загальний контакт, до якого підключаються всі негативні полюси світлодіодів (аноди).



Рисунок 1.32 – RGB світлодіод

Для кожного світлодіода потрібен власний резистор на 270 Ом, щоб запобігти можливості протікання надто великих струмів. Ці резистори



встановлюються в коло між катодом (червоний, зелений і синій) і керуючими пинами на нашому Arduino.

Рисунок 1.33 – Умовне позначення світлодіодів на схемі

Також існують RGB світлодіоди із загальним анодом, замість загального катода, найдовший контакт на світлодіоді підключається до виводу +5 V замість землі.

Принцип дії RGB світлодіода

Ми можемо змішувати три основні кольори і бачити нові відтінки, так як в наших очах три типи рецепторів (для червоного, зеленого і синього кольорів). В

результаті наше око і мозок обробляє інформацію про насиченість цих трьох кольорів і перетворює їх в інші відтінки спектру [24].

Тобто, використовуючи одночасно три світлодіода, ми немов обманюємо наші очі. Ця ж ідея використовується в телевізорах, де рідкокристалічний дисплей складається з маленьких точок червоного, зеленого і синього кольорів, які розташовані дуже близько один до одного і формують окремі пікселі.

Якщо ми налаштуємо однакову яскравість всіх світлодіодів, при цьому він буде світитися білим. Якщо ми відключимо синій світлодіод і будуть горіти з однаковою яскравістю тільки червоний і зелений, ми отримаємо жовте світло [25].

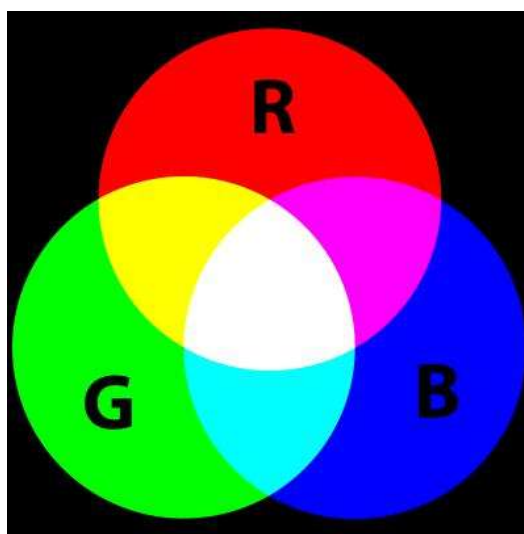


Рисунок 1.34 – Принцип змішування кольорів

Можна управляти яскравістю кожного світлодіода окремо, змішуючи кольори. Так як чорний колір не що інше, як відсутність світла, отримати його не вийде. Найближчий відтінок чорного - це повністю вимкнені світлодіоди.

Схема підключення світлодіода до Ардуіно зображена на рисунку 1.35

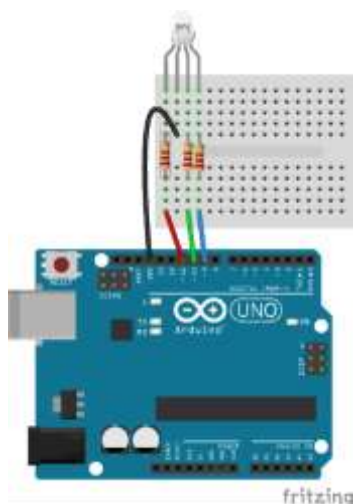


Рисунок 1.35 - Схема підключення RGB світлодіода до Ардуіно

1.3.2.6. Модуль звуку

Модуль звуку для Ардуіно має назву зумер, пьезодінамік або навіть баззер. Дані модулі використовуються для звукового оповіщення в тих пристроях і системах, для функціонування яких в обов'язковому порядку потрібний звуковий сигнал. Широко поширені зумери в різній побутовій техніці та іграшках, які використовують електронні плати. Пьезодінамік перетворюють команди, засновані на двохбітній системі числення 1 і 0, в звукові сигнали.



Рисунок 1.36 - П'єзoeлемент або зуммер

Зуммер конструктивно представлений металевою пластиною з нанесенням на неї напилення з струмопровідної кераміки. Пластина і напилення виступають в ролі контактів. Пристрій полярно, має свої «+» і «-». Принцип дії зумера

заснований на відкритому братами Кюрі в кінці дев'ятнадцятого століття п'єзоелектричному ефекту. Згідно з ним, при подачі електрики на зумер він починає деформуватися. При цьому відбуваються удари об металеву пластинку, яка і виробляє "шум" потрібної частоти [26].

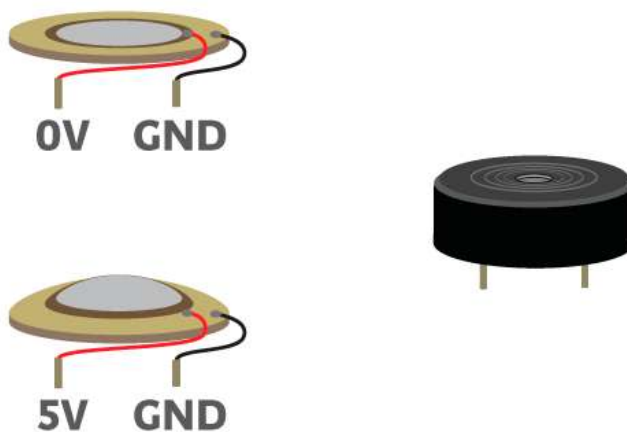


Рисунок 1.37 – Принцип дії зуммера

Зумер буває двох видів: активний і пасивний. Принцип дії у них однаковий, але в активному немає можливості змінювати частоту звучання, хоча сам звук голосніше і підключення простіше.

Конструктивно модуль виповнюється в самих різних варіантах. Самий рекомендований для підключення до Ардуїно - готовий модуль з вбудованою обв'язкою. Приклад такого модуля зображено на рисунку 1.38.



Рисунок 1.38 – Модуль звуку для Ардуїно

Якщо порівнювати зі звичайними електромагнітними перетворювачами звуку, то зуммер має більш просту конструкцію, що робить його використання

економічно обґрунтованим. Частота одержуваного звуку задається користувачем в програмному забезпеченні.

Відмінності активного і пасивного зумера

Головна відмінність активного зумера від пасивного полягає в тому, що активний зумер генерує звук самостійно. Для цього користувач повинен просто включити або виключити його, іншими словами, подавши напругу на контакти або знеструмивши. Пасивний зумер ж вимагає джерела сигналу, який задає параметри звукового сигналу. В якості такого джерела може виступати плата Ардуіно. Активний зумер буде видавати більш гучний звуковий сигнал в порівнянні з його конкурентом. Частота випромінюваного звуку активного зумера становить значення $2,5 \text{ кГц} \pm 300 \text{ Гц}$. Напруга живлення для зуммера варіюється від 3,5 до 5 В.

Активний пьезодінамік краще ще через те, що в скетчі не потрібно створювати додатковий фрагмент коду з затримкою, що впливає на робочий процес. Також для визначення того, що за елемент знаходиться перед користувачем, можна виміряти опір між двома проводами. Більш високі значення будуть вказувати на активний зумер Ардуіно.

Підключення зуммера до Ардуіно зображено на рисунку 1.39.

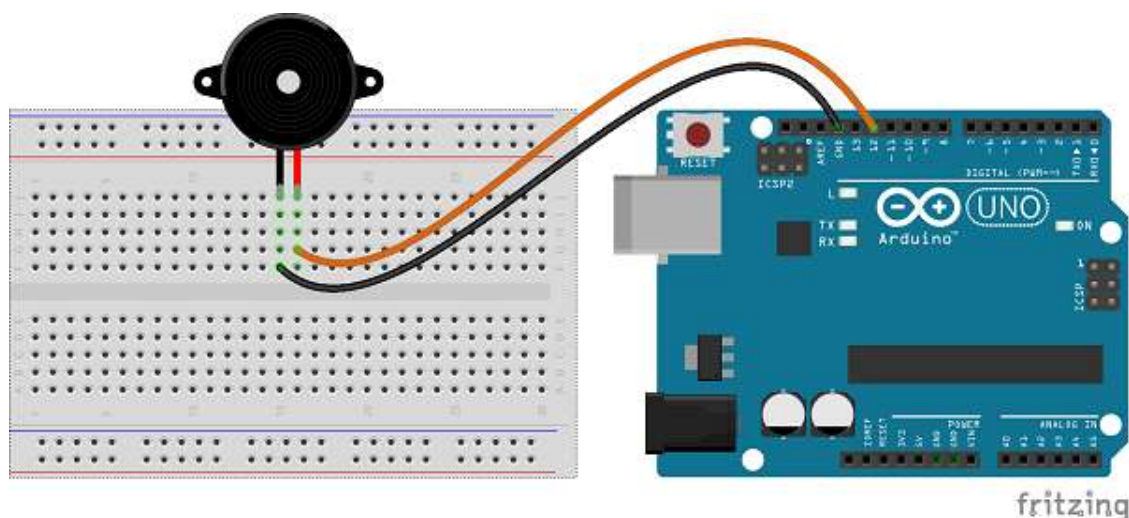


Рисунок 1.39 – Підключення зуммера до Ардуіно

Електрична схема підключення п'єзоелемента без супроводжуючих модулів зображена на рисунку 1.40.

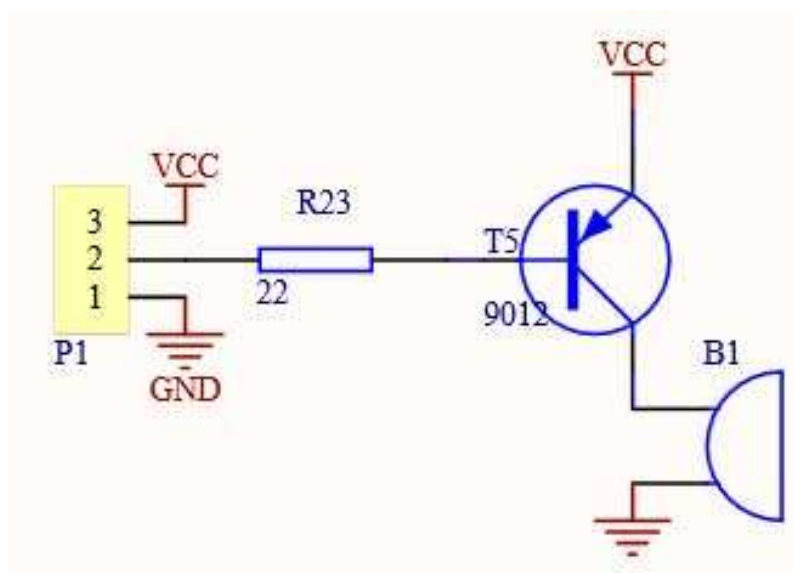


Рисунок 1.40 - Схема підключення п'єзoeлементу без супроводжуючих модулів

Зумер має два виходи. Слід звернути увагу на їх полярність. Темний провід повинен бути підключений до «землі», червоний - до цифрового входу з ШІМ. Один вивід налаштовується в програмі як «вхід». Arduino відстежує коливання напруги на виводі, на який подається напруга з кнопки, резистора або датчиків.

1.3.2.7. Дисплей LCD 1602

Рідкокристалічний дисплей (Liquid Crystal Display [27]) LCD 1602 потрібен для виведення рядків символів в різних проектах. Він коштує недорого, є різні модифікації з різними кольорами підсвічування, ви можете легко завантажити готові бібліотеки для скетчів Ардуіно. Але найголовнішим недоліком цього екрану є той факт, що дисплей має 16 цифрових виводів, з яких обов'язковими є мінімум 6. Тому використання цього LCD екрана без I2C додає обмеження для плат Arduino Uno або Nano.

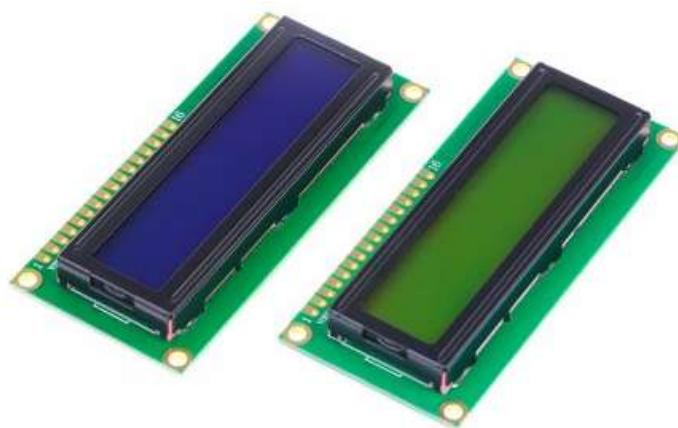


Рисунок 1.41 - Дисплей LCD 1602

Кожен з виводів має своє призначення:

1. Земля GND;
2. Живлення 5 В;
3. Установка контрастності монітора;
4. Команда, дані;
5. Записування і читання даних;
6. Enable;
- 7-14. Лінії даних;
15. Плюс підсвічування;
16. Мінус підсвічування.



Рисунок 1.42 – Виводи LCD 1602

Технічні характеристики дисплея:

- символний тип відображення, є можливість завантаження символів;
- світлодіодне підсвічування;
- контролер HD44780;
- напруга живлення 5В;
- формат 16x2 символів;
- діапазон робочих температур від -20С до + 70С;
- діапазон температур зберігання від -30С до +80 С;
- кут огляду 180 градусів.

Стандартна схема приєднання монітора безпосередньо до мікроконтролера Ардуіно без I2C зображена на рисунку 1.43.

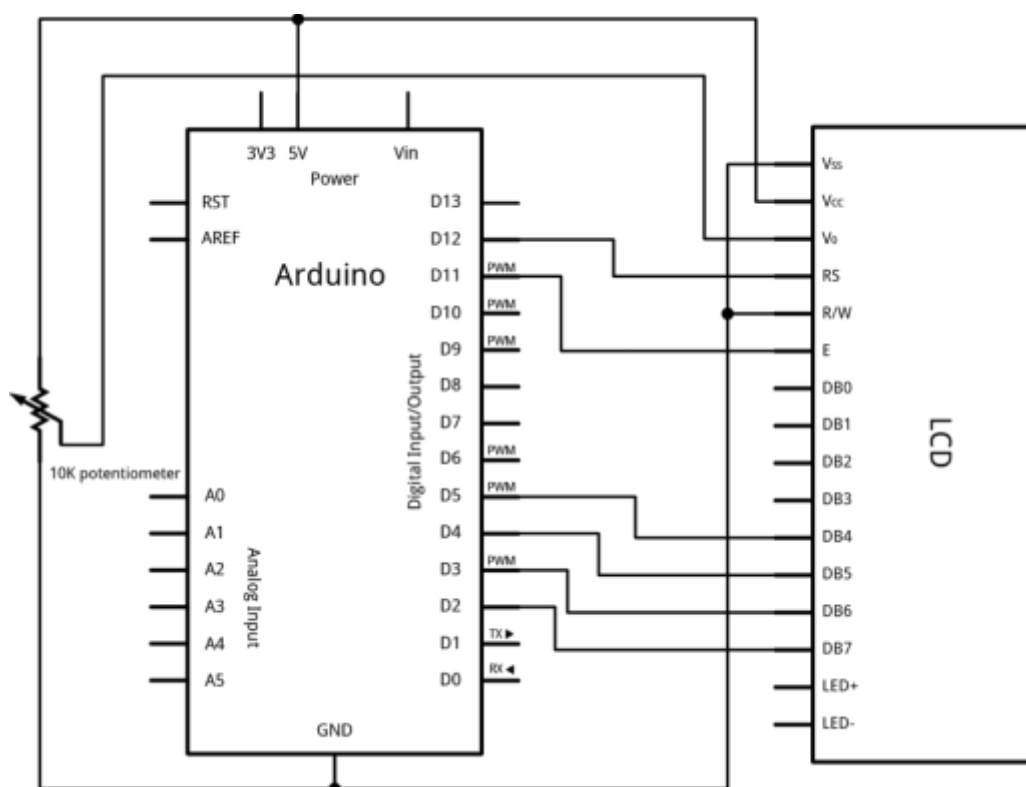


Рисунок 1.43 - Схема підключення дисплея до Ардуіно без I2C

Через велику кількість контактів може не вистачити місця для приєднання інших периферійних пристроїв. Використання I2C зменшує кількість проводів до 4, а зайнятих виводів мікроконтролера до 2.

В роботі I2C можна виділити кілька переваг:

- для роботи потрібно всього 2 лінії - SDA (лінія даних) і SCL (лінія синхронізації);
- підключення великої кількості провідних приладів;
- зменшення часу розробки;
- для управління всім набором пристроїв потрібно тільки один мікроконтролер;
- можливе число під'єднаних мікросхем до однієї шини обмежується тільки граничною ємністю;
- шина вже інтегрована в саму Arduino, тому не потрібно розробляти додатково шинний інтерфейс.

Недоліки:

- існує ємнісне обмеження на лінії - 400 пФ;
- важке програмування контролера I2C, якщо на шині є кілька різних пристроїв;
- при великій кількості пристроїв виникає труднощі локалізації збою, якщо одне з них помилково встановлює стан низького рівня.



Рисунок 1.44 - Модуль i2c для LCD 1602

Найшвидший і зручний спосіб використання i2c дисплея в Ардуіно - це готовий екран з вбудованою підтримкою протоколу.

Для підключення до плати Ардуіно використовуються i2c виходи. Якщо потрібно, підключаємо до зовнішнього джерела живлення для підсвітки. За

допомогою вбудованого підлаштування резистора ми можемо налаштувати установлювані значення контрастності.

Підключення LCD екрану до Ардуіно по I2C зображено на рисунку 1.45

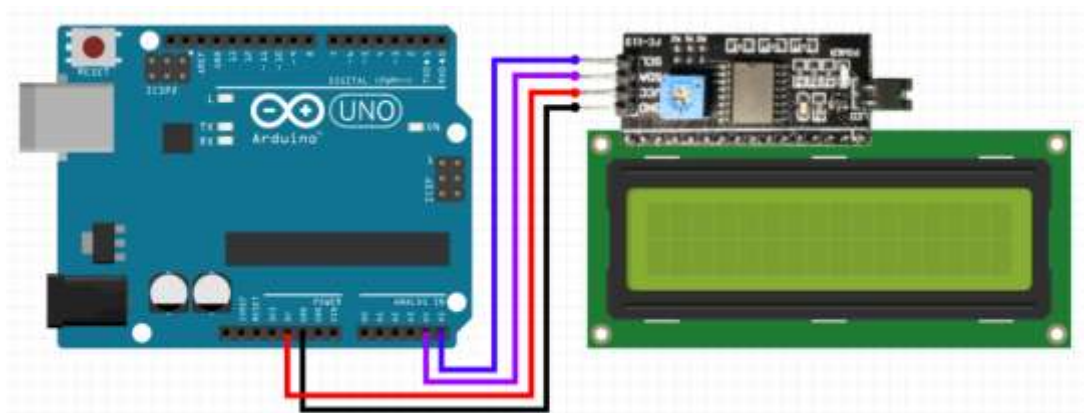


Рисунок 1.45 - Підключення LCD екрану до Ардуіно по I2C

Рідкокристалічний монітор з підтримкою i2c підключається до плати за допомогою чотирьох проводів - два дроти для даних, два дроти для живлення.

- Висновок GND підключається до GND на платі.
- Висновок VCC - на 5V.
- SCL підключається до піну A5.
- SDA підключається до піну A4.

1.3.3. Пристрої вводу та виводу інформації - модуль Bluetooth

Модуль Bluetooth [28] спілкується з Arduino за допомогою послідовного з'єднання. Він має чотири виводи, які зображені на рисунку 1.46:



Рисунок 1.46 - Модуль Bluetooth

- VCC - підключається до виводу 5V;
- GND - контакт заземлення;
- TXD використовується для передачі даних від модуля до Arduino. Він повинен бути підключений до виводу RX в Arduino, яка є контактом 0 у випадку Uno;
- RXD використовується для отримання даних з Arduino. Повинен бути підключений до виводу TX Arduino, який є контактом 1 у разі Arduino Uno.

На модулі, вхід TX розрахований на 3,3 В. Це означає, що, хоча ми можемо жити модуль від 5 В з Arduino, лінії зв'язку від і до модуля повинні бути в 3,3 В.

Для підключення треба використати два резистори для створення подільника напруги і зменшення напруги сигнального виводу TX Arduino від 5 В до приблизно 3,3 В. Схема підключення модуля Bluetooth до Ардуіно зображено на рисунку 1.47.

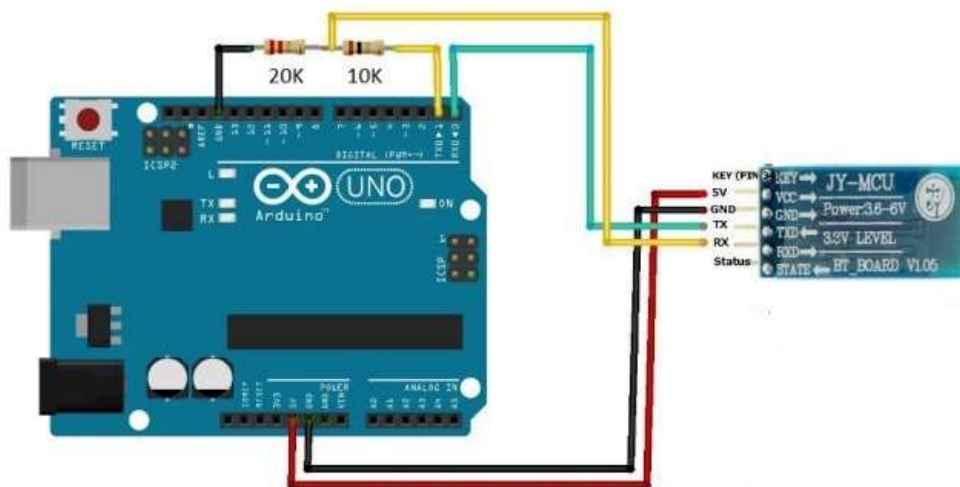
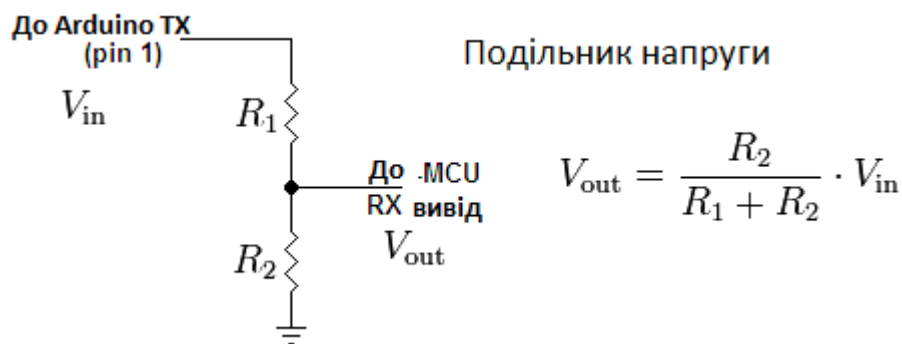


Рисунок 1.47 - Схема підключення модуля Bluetooth до Ардуіно

На схемі використані резистори 20кОм і 10 кОм, на основі рівняння для подільника напруги:



$R_1 = 10\text{кОм}$, $R_2 = 20\text{кОм}$, $V_{in} = 5\text{ В}$. Для V_{out} отримуємо: $V_{out} = 5\text{В} \times 20\text{кОм} / (20\text{кОм} + 10\text{кОм}) = 5\text{В} \times 20\text{кОм} / 30\text{кОм} = 5\text{В} \times 2/3 = 10\text{В}/3 = 3.33\text{ В}$.

Можна використовувати будь-яку іншу комбінацію резисторів, але R_2 повинно мати в два рази менший опір, ніж R_1 .

Висновки до розділу

В розділі розглянуто історію мікропроцесорної техніки, зроблено огляд мікроконтролерів Arduino Uno та Arduino Mega для системи контролюваного нагрівання води. Наведено опис периферійних пристроїв, які використовуються в цій системі. А також зроблено короткий огляд кожного з них та описано принципи їх дії.

2 ТЕОРЕТИЧНІ ТА ПРАКТИЧНІ ЗАСАДИ РОЗРОБКИ ПРИСТРОЮ

Розробка методів, які дозволять пристрою виконувати алгоритми для функціонування системи розроблено в середовище розробки Arduino IDE [29].

Абревіатура IDE розшифровується як Integrated Development Environment, в перекладі - інтегроване середовище розробки. З її допомогою скетч, написаний на мові Arduino, перевіряється, перетворюється в C++, компілюється, завантажується в Arduino. Для збільшення можливостей функціонування пристрою, доречно застосовувати додаткові бібліотеки Ардуіно.

2.1. Додаткові бібліотеки для розробки коду

Бібліотека Ардуіно - це програмний код, що зберігається не в скетчі, а в зовнішніх файлах, які можна підключити до вашого проекту [30]. У бібліотеці зберігаються різні методи і структури даних, які потрібні для спрощення роботи з датчиками, індикаторами, модулями та іншими компонентами. Використання готових програм істотно спрощує роботу над проектами.

Існує велика кількість бібліотек, які можна застосувати в різних проектах. Переважна більшість бібліотек поширюються за вільною ліцензією. Для програмування пристрою контрольованого нагрівання води потрібні такі додаткові бібліотеки, як: LiquidCrystal_I2C.h, OneWire.h, ServoSmooth.h, GyverButton.h.

2.1.1. LiquidCrystal_I2C.h.

Для роботи з символьними графічними дисплеями потрібна бібліотека LiquidCrystal_I2C [31] яка входить в стандартний набір Arduino IDE і призначена для роботи символьного дисплея по апаратної шині I2C.

Функції для бібліотеки LiquidCrystal_I2C:

- `begin (cols, rows, [char_size]);` - Ініціалізація дисплея із зазначенням кількості стовпців, рядків і розміру символу;
- `clear ();` - Очищення дисплея з установкою курсору в положення 0,0;
- `home ();` - Установка курсору в положення 0,0;
- `display ();` - Швидке включення дисплея (без зміни даних в ОЗУ);
- `noDisplay ();` - Швидке вимикання дисплея (без зміни даних в ОЗУ);
- `blink ();` - Включення миготливого курсору (з частотою близько 1 Гц);
- `noBlink ();` - Виключення миготливого курсору;
- `cursor ();` - Включення підкреслення курсору;
- `noCursor ();` - Виключення підкреслення курсору;
- `scrollDisplayLeft ();` - Прокрутка дисплея вліво. Зрушення координат дисплея на один стовпець вліво (без зміни ОЗУ);
- `scrollDisplayRight ();` - Прокрутка дисплея вправо. Зрушення координат дисплея на один стовпець вправо (без зміни ОЗУ);
- `leftToRight ();` - Вказує надалі зрушувати становище курсору, після виведення чергового символу, на один стовпець вправо;
- `rightToLeft ();` - Вказує надалі зрушувати становище курсору, після виведення чергового символу, на один стовпець вліво;
- `noAutoscroll ();` - Вказує надалі вирівнювати текст по лівому краю від позиції курсора;
- `autoscroll ();` - Вказує надалі вирівнювати текст по правому краю від позиції курсора;
- `createChar (num, array);` - Запис призначеного для користувача символу в CGRAM дисплея під зазначеним номером;
- `setCursor (col, row);` - Установка курсору в позицію зазначену номером колонки і рядка;
- `print (text);` - Виведення тексту, символів або цифр на екран дисплея. Синтаксис схожий з однойменною функцією класу `Serial`;
- `init ();` - Ініціалізація дисплея. Повинна бути першою командою бібліотеки `LiquidCrystal_I2C` після створення об'єкта;

- `backlight ()`; - Включення підсвічування дисплея;
- `noBacklight ()`; - Вимкнення підсвічування дисплея;
- `setBacklight (flag)`; - Управління підсвічуванням (`true` - включити / `false` - вимкнути), використовується замість функцій `noBacklight` і `backlight`.

Лістинг програми для підключення по шині I2C:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (address, col, row);
void setup () {
  lcd.init (); }
```

Де параметр:

- `address`: Адреса дисплея на шині I2C - `0x27` або `0x3F`;
- `col`: кількість стовпців, які реалізовані у дисплея;
- `row`: кількість рядків, які реалізовані у дисплея.

2.1.2. OneWire.h

1-Wire – це однопровідний інтерфейс [32]. Для обміну даними з пристроями з цього інтерфейсу потрібно всього один сигнальний провід і один провід «землі». Пристрої спроектовані таким чином, що з цього ж сигнального проводу вони живляться (тобто для живлення пристрою використовується сигнальний провід), такий режим називається «паразитне живлення». Для реалізації такого способу живлення в пристроях ставлять досить ємний конденсатор.

Для того щоб почати сесію обміну даними, необхідно сформувати сигнал «скидання». Для цього майстер передає в лінію даних 0 на час не менше 480 мкс. Після чого майстер відпускає лінію даних і починає «слухати» лінію. За рахунок резистора підтяжки лінія даних встановиться в 1. Якщо на лінії присутній пристрій (датчик), то він передає майстру підтвердження скидання, для цього він утримує лінію даних в 0 на час 60-240 мкс. Прочитавши стан лінії, майстер дізнається про присутність на шині пристроїв, готових до обміну.

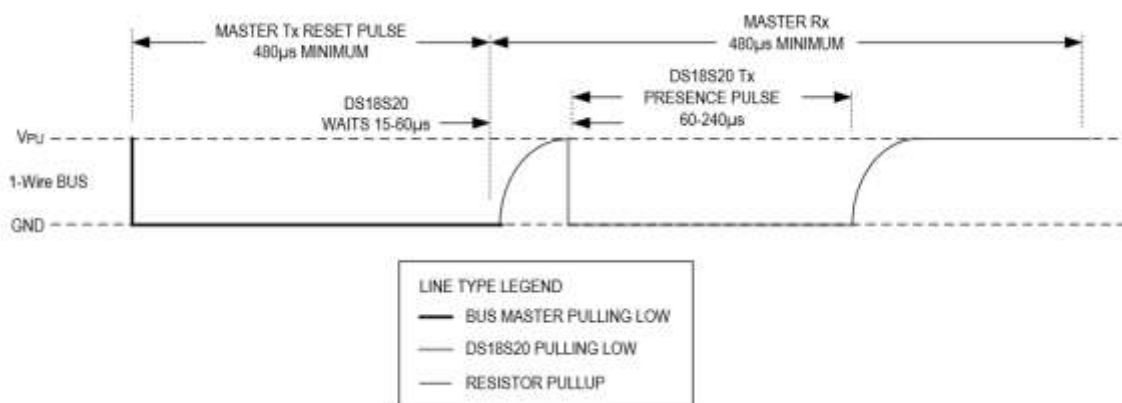


Рисунок 2.1 - Старт обміну даними [32]

1-Wire володіє ще однією особливістю: передача бітів здійснюється не рівнями сигналу, а тимчасовими затримками. Таким чином, щоб передати 1, необхідно встановити в лінії 0 і тримати його 15 мкс, після чого відпустити лінію, яка за рахунок підтягувального резистора перейде в рівень 1. Щоб передати 0, необхідно встановити в лінії 0 на 15 мкс, а потім тримати 0 на лінії ще 60-120 мкс.

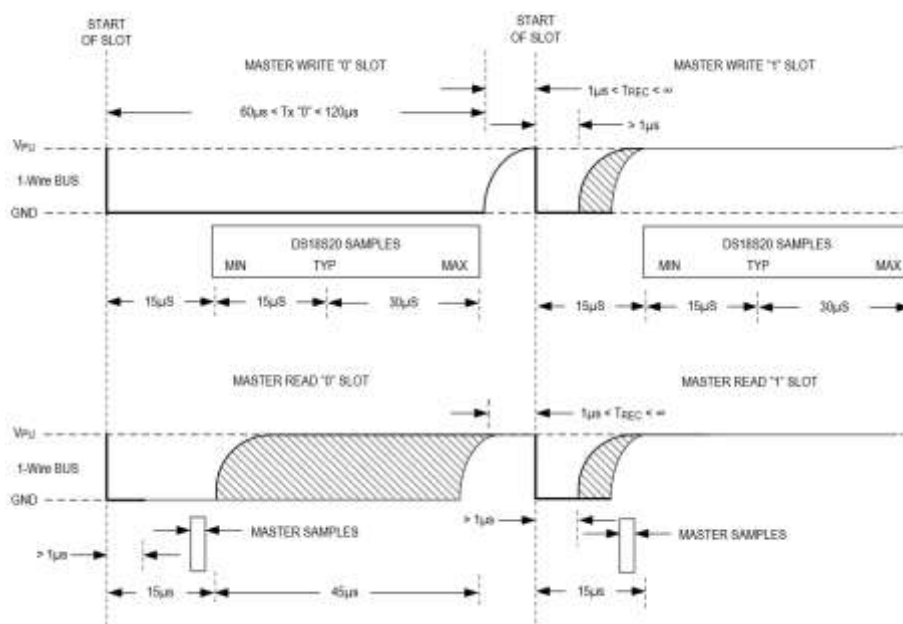


Рисунок 2.2 - Передача даних 1-Wire [32]

2.1.3. ServoSmooth.h

Дана бібліотека є "додатком" до стандартної бібліотеки Servo.h [33] і дозволяє плавно керувати сервоприводом. Суть роботи полягає в методі `tick()`, який потрібно викликати постійно в `loop`, всередині тика знаходиться алгоритм з

власним таймером, який трошки повертає серво до потрібного стану. Бібліотека дублює кілька методів з Servo.h (attach має розширену ініціалізацію):

- write () і writeMicroseconds () - повернуть вал серво з максимальною швидкістю;

- attach () і detach () - підключити і відключити серво від управління.

Лістинг програми для ініціалізації бібліотеки:

```
#include "ServoSmooth.h" // підключаємо бібліотеку
```

```
ServoSmooth servo; // створюємо об'єкт
```

За ініціалізації (attach ()) є кілька варіантів:

- attach (pin) - підключить серво на вказаний pin, кут повороту буде встановлений на 0 градусів. Довжина імпульсу буде стандартна, 500-2400 мкс;

- attach (pin, target) - підключить серво на вказаний pin, кут повороту буде встановлений на target градусів. Довжина імпульсу буде стандартна, 500-2400 мкс;

- attach (pin, min, max) - підключить серво на вказаний pin, кут повороту буде встановлений на 0 градусів. Довжина імпульсу буде встановлена min і max відповідно;

- attach (pin, min, max, target) - підключить серво на вказаний pin, кут повороту буде встановлений на target градусів. Довжина імпульсу буде встановлена min і max відповідно.

Сервопривід управляється ШІМ сигналом, в якому довжина імпульсу прямо керує кутом повороту, тобто подаючи мінімальну і максимальну довжину ми отримуємо робочий кут 180 градусів. За замовчуванням довжина імпульсу встановлені 500 і 2400 відповідно, що підходить більшості сервоприводів, але бажано подивитися і відкалібрувати свій привід так, щоб він працював на всі 180 градусів.

Керування сервоприводом

Рух сервоп відбувається автоматично в методі tick(), потрібно всього лише викликати його як можна частіше в loop (tick() має вбудований таймер на 20 мілісекунд). Також є метод tickManual(), який повертає сервопривід на наступний

«крок» при кожному виклику (той же `tick()`, але не має свого таймера). Обидва методи `tick()` повертають `false`, поки сервопривід рухається, і `true`, коли сервопривід досяг встановленого кута. Також сервопривід автоматично відключається від управління при досягненні заданого кута повороту. Цю функцію можна вимкнути, викликавши `setAutoDetach (false)`. Інструменти для управління рухом приводу:

- `setTarget` (довжина) - встановлює цільову позицію для серво в величині довжина імпульсу, мкс (~ 500-2400);
- `setTargetDeg` (кут) - встановлює цільову позицію для серво в градусах (0-180);
- `setSpeed` (швидкість) - установка максимальної швидкості (умовні одиниці, 0 - 200);
- `setAccel` (прискорення) - установка прискорення (0.05 - 1). При значенні 1 прискорення максимальне;
- `start ()` - автоматичний `attach` + дозволяє роботу `tick` - серво рухається до заданої позиції;
- `stop ()` - `detach` + забороняє роботу `tick` - серво зупиняється;

Корисні допоміжні методи для різних ситуацій:

- `setCurrent` (довжина) - установка поточної позиції в мкс (500 - 2400);
- `setCurrentDeg` (кут) - установка поточної позиції в градусах (0-180). Залежить від `min` і `max`;
- `getCurrent ()` - отримання поточної позиції в мкс (500 - 2400);
- `getCurrentDeg ()` - отримання поточної позиції в градусах (0-180). Залежить від `min` і `max`.

2.1.4. GyverButton.h

Бібліотека `GyverButton.h` [34] створена для максимально простого і зручного одержання всіх можливостей звичайної тактової кнопки: клік, утримання, кілька

натискань і зміна змінної при утриманні з кроком по часу. У десятки разів спростить і скоротить лістинг програми.

- Робота з нормально замкнутими і нормально розімкнутими кнопками;
- Робота з підключенням PULL_UP і PULL_DOWN;
- Опитування кнопки з програмним усуненням брязкіту контактів;
- Настроюється час антибрязкіту;
- Відпрацювання натискання, утримування, відпускання, кліка по кнопці;
- Настроюваний таймаут утримання;
- Відпрацювання одиночного, подвійного і потрійного натискання;
- Відпрацювання будь-якої кількості натискань кнопки;
- Настроюваний таймаут повторного натискання / утримання;
- Функція зміни значення змінної з заданим кроком і через встановлені проміжки часу за часом.

Функції і методи C ++:

GButton (uint8_t pin); // клас кнопки, приймає пін

GButton (uint8_t PIN, boolean type, boolean dir); // клас кнопки, приймає PIN пін, тип type (HIGH_PULL / LOW_PULL) і напрямок dir (NORM_OPEN / NORM_CLOSE), де:

- HIGH_PULL - кнопка підключена до GND, пін підтягнутий до VCC, pinMode - INPUT_PULLUP (за замовчуванням)

- LOW_PULL - кнопка підключена до VCC, пін підтягнутий до GND, pinMode - INPUT

- NORM_OPEN - кнопка за замовчуванням розімкнута

- NORM_CLOSE - кнопка за замовчуванням замкнута

void setDebounce (uint16_t debounce); // установка часу антибрязкіту (за замовчуванням 80 мс)

void setTimeout (uint16_t timeout); // установка таймаута утримання (за замовчуванням 500 мс)

void setStepTimeout (uint16_t step_timeout); // установка таймаута між інкрементами (за замовчуванням 400 мс)

`void setType (boolean type);` // установка типу кнопки (HIGH_PULL - підтягнута до живлення, LOW_PULL – стянутий до землі)

`void setTickMode (boolean tickMode);` // (MANUAL / AUTO) ручний або автоматичний опитування кнопки функцією `tick ()`, де:

- MANUAL - потрібно викликати функцію `tick ()` вручну;
- AUTO - `tick ()` входить в усі інші функції і опитується сама.

`void tick ();` // опитування кнопки

`void tick (boolean state);` // опитування зовнішнього значення

`boolean isPress ();` // повертає true при натисканні на кнопку

`boolean isRelease ();` // повертає true при відпуску кнопки

`boolean isClick ();` // повертає true при кліці

`boolean isHoled ();` // повертає true при утриманні довше timeout

`boolean isHold ();` // повертає true, утримуючи кнопку, не скидається

`boolean state ();` // повертає стан кнопки

`boolean isSingle ();` // повертає true при одиночному натисканні

`boolean isDouble ();` // повертає true при подвійному натисканні

`boolean isTriple ();` // повертає true при потрібному кліку

`boolean hasClicks ();` // перевірка на наявність кліків

`uint8_t getClicks ();` // повернути кількість кліків

`boolean isStep ();` // повертає true за таймером `setStepTimeout`

2.2. Види інтерфейсів для взаємодії периферійних пристроїв

Послідовний зв'язок є найбільш широко використовуваним комунікаційною методологією для роботи периферійних пристроїв із мікроконтролером. Дані зв'язку передаються послідовно (один за іншим), а не паралельно (всі разом). Таким чином, послідовний зв'язок реалізовано з використанням меншої кількості проводів в порівнянні з паралельним методом комунікації, при цьому також потрібен якийсь механізм синхронізації для успішного зв'язку. В разі послідовних

протоколів зв'язку тільки сама передача даних виконується послідовно, все інше, обробка даних, тощо виконується паралельно, тобто у вигляді регістрів.

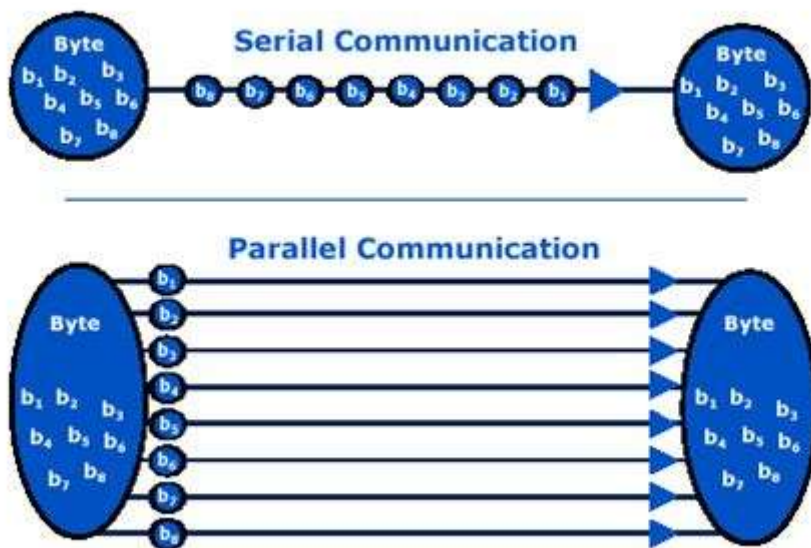


Рисунок 2.3 - Послідовний та паралельний зв'язок [35]

Послідовний зв'язок класифікується на синхронний і асинхронний тип зв'язку [35]. При синхронному типі передавач і приймач мають загальну систему тактування, щоб синхронізуватися між собою. Асинхронний тип послідовного зв'язку не потребує ніякого загального джерела синхронізації між передавачем і приймачем, обидві сторони працюють відповідно до їх незалежних систем тактування.

2.2.1. SPI (Serial Peripheral Interface)

Це протокол послідовного зв'язку синхронного типу, який складається з двох ліній даних (MOSI і MISO), однієї тактовою лінії (SCK) і лінії вибору підлеглих (SS) [36], де:

Master - пристрій, що забезпечує синхронізацію

Slave - пристрій, що використовує тактування мастера для зв'язку

MOSI - Master Out Slave In (лінія, по якій майстер відправляє дані)

MISO - Master In Slave Out (лінія, по якій слейв відправляє дані у відповідь)

SCK - лінія тактування (надається провідним пристроєм)

SS - Slave Select (лінія, яка використовується для вибору пристрою, до якого треба встановити зв'язок)

У випадку з SPI в будь-який момент часу може бути тільки один slave пристрій і кілька інших slave пристроїв, які відповідають тільки на виклик master пристрою. Весь зв'язок обробляється master пристроєм; жоден slave пристрій не може відправляти дані самостійно. Master відправляє дані через MOSI, а slave пристрої відповідають через лінію MISO. У всьому процесі SCK (послідовне тактування) грає дуже важливу роль, щоб читати дані з MOSI і відповідати через MISO. SS (вибір slave пристрою) використовується для того, щоб конкретний slave пристрій прокинувся. Принцип підключення за допомогою інтерфейсу SPI наведено на рисунку 2.4.

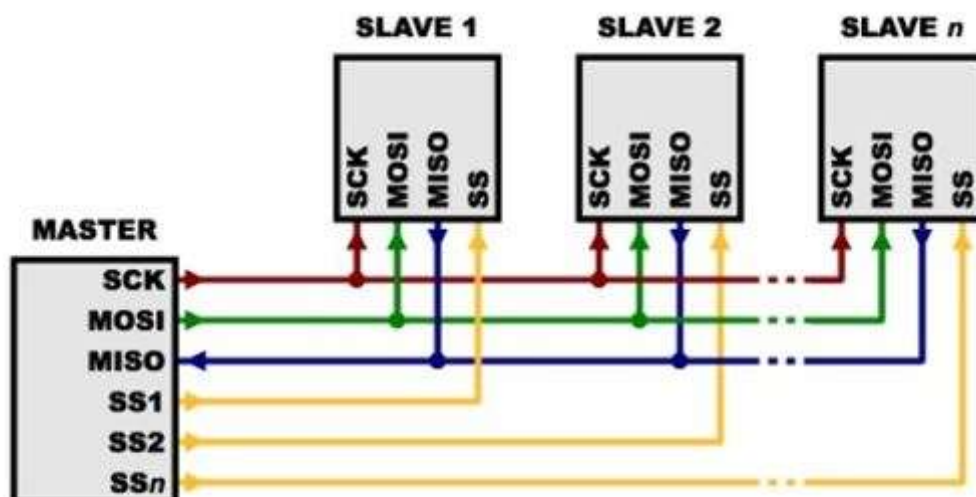


Рисунок 2.4 - Принцип підключення інтерфейсу SPI [36]

Існує декілька регістрів, які використовуються для реалізації зв'язку SPI. Всі регістри мають довжину 8 біт:

SPDR (реєстр даних SPI) використовується для зберігання одного байта даних, який повинен бути переданий або отриманий.

SPSR (реєстр стану SPI) містить біти стану, які беруть участь у передачі SPI.

SPCR (реєстр управління SPI) містить контрольні біти, які беруть участь у передачі SPI.

Переваги інтерфейсу SPI:

- він забезпечує синхронний послідовний зв'язок, який надійніше асинхронного;
- декілька slave пристроїв можуть бути підключені до одного master пристрою;
- швидка форма послідовного зв'язку.

Недоліки:

- потрібно кілька ліній вибору slave пристроїв для підключення декількох slave пристроїв;
- тільки master пристрій контролює весь процес комунікації; ніякі slave пристрої не можуть безпосередньо зв'язуватися один з одним.

2.2.2. I2C (Inter-Integrated Circuit) або двопровідний інтерфейс

Іншим дуже корисним синхронним протоколом послідовної зв'язку є протокол I2C або Inter-Integrated Circuit [37]. На відміну від SPI, I2C використовує тільки два дроти для всього процесу, тому він також відомий як протокол двухпроводного інтерфейсу (TWI). Протокол I2C може підтримувати декілька периферійних пристроїв, але на відміну від SPI, який підтримує тільки один провідний пристрій, I2C може також підтримувати декілька майстер-пристроїв. Кожен пристрій відправляє / приймає дані, використовуючи тільки один провід SDA. А провід SCL підтримує синхронізацію між пристроями через загальну систему тактування, яка реалізується активним ведучим пристроєм.

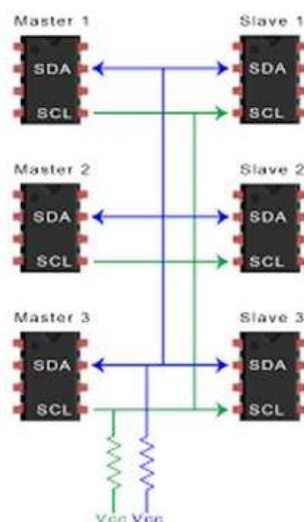


Рисунок 2.5 – Принцип взаємодії I2C інтерфейсу

Кожен пристрій має свій власний унікальний адрес (від 7 до 10 біт), який веде використовує для їх ідентифікації. Кожне повідомлення між пристроями починається з умови запуску і закінчується умовою зупинки. Одне повідомлення може містити кілька байтів даних, кожен з яких має біт підтвердження (ACK) або негативний біт підтвердження (NACK) між ними.

Переваги:

- пристрої можуть з'єднуватися один з одним;
- для цього протоколу потрібно тільки два дроти.

Недоліки:

- протоколу I2C повільніше в порівнянні з SPI, тому що в рамках цього протоколу виконується багато операцій з кадрами даних.

2.2.3. UART

UART - це універсальний асинхронний приймач і передавач [38], а USART - універсальний синхронний і асинхронний приймач і передавач. Різниця між ними полягає в тому, що UART виконує тільки асинхронний послідовний зв'язок, в той час як USART може виконувати як синхронний, так і асинхронний послідовний комунікаційний процес.

Для асинхронного режиму цей протокол використовує тільки два дроти, а саме Rx і Tx. Оскільки тут не потрібна синхронізація, обидва пристрої повинні використовувати свої незалежні внутрішні системи тактування для функціонування. Швидкість передачі допомагає цим пристроям залишатися в режимі синхронізації, фіксуючи швидкість обміну даними. Швидкість передачі даних в бодах – це одно число біт даних, що передається в секунду, тому обидва пристрої повинні працювати з однаковою швидкістю передачі в бодах.

Інтерфейс UART / USART має велике обмеження, пов'язане з тим, що тільки два пристрої можуть обмінюватися даними за допомогою цього протоколу одночасно. Лінія Tx одного пристрою передає дані на лінію Rx іншого пристрою і аналогічно Tx останнього передає дані в Rx першого пристрою. Так відбувається обмін даними.

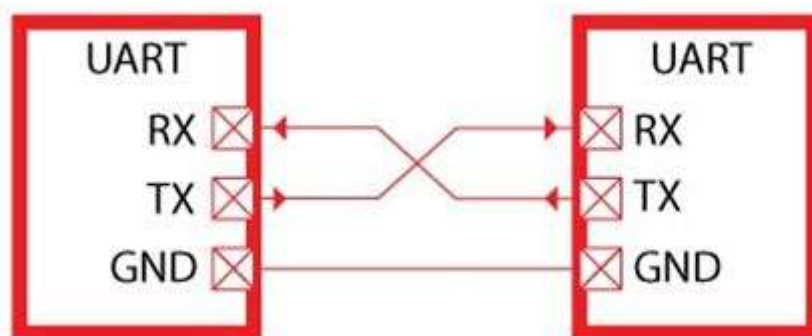


Рисунок 2.6 – Принцип взаємодії UART інтерфейсу

Для синхронного режиму використовується додатковий вивід XCK. Імпульси тактування генеруються пристроєм, які відправляють дані в цей час.

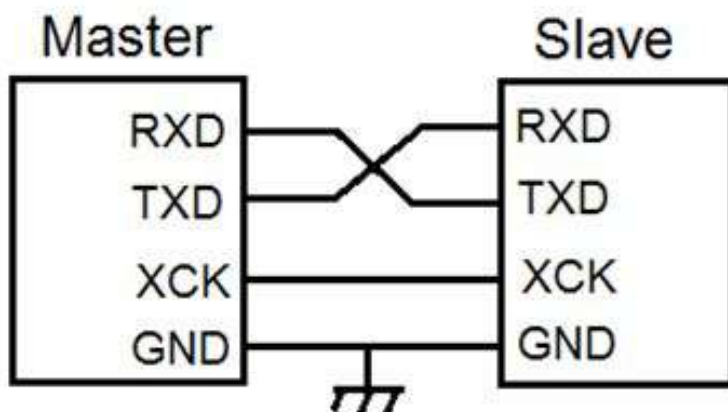


Рисунок 2.7 – UART інтерфейсу в синхронному режимі

Переваги UART / USART:

- протокол забезпечує як синхронну, так і асинхронну послідовну зв'язок;
- забезпечується наявність різних швидкостей передачі, що робить його придатним для широкого застосування і пристроїв;
- це одна з найпростіших форм послідовної зв'язку.

Недолік:

- одночасно можуть бути підключені тільки два пристрої.

2.3 Вдосконалення мікроконтролера Arduino Uno

Обраний мікроконтролер Arduino Uno має обмежену кількість входів для зчитування сигналу й виводів для відправки. Тому доречно вдосконалити мікроконтролер так, щоб збільшити кількість ввідів та виводів, підключити більшу кількість кнопок для комутації сигналу, а також усунути брязкіт контактів.

2.3.1 Брязкіт контактів - програмне і апаратне усунення

Брязкіт контактів (англ. *bounce* - брязкіт, *debounce* - усунення брязкоту) - явище, що має місце в електромеханічних комутаційних пристроях, триває деякий час після замикання електричних контактів і складено з багаторазових неконтрольованих замикань і розмикань контактів, обумовлених пружністю матеріалів та деталей контактної системи. Впродовж деякого часу контакти «підстрибують» під час зіткнень, «спотворює» електричне коло.

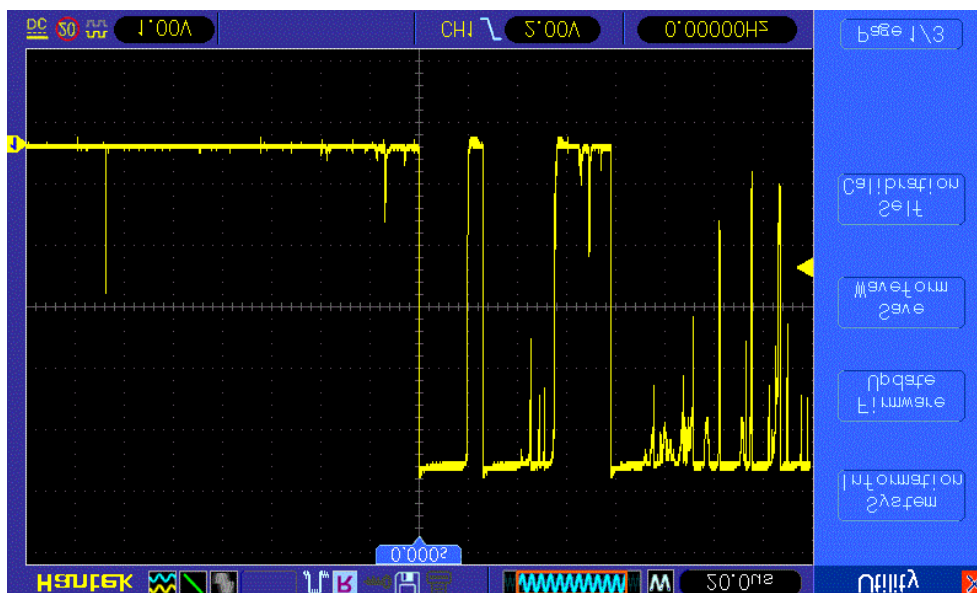


Рисунок 2.8 - Брязкіт під час натискання кнопки, зафіксований осцилографом

Це негативне явище можна усунути двома шляхами - програмно і апаратно.

Програмне усунення брязкоту.

Для програмного усунення брязкоту, у середовищі з Arduino IDE після першої зміни стану кнопки, треба встановити паузу в декілька мілісекунд, для того щоб контакт стабілізувався і брязкіту не було. Наведено лістинг функції *debounce*:

```
bool debounce () {
    bool current = digitalRead (pinButton);
    if (current != lastButton) { // Старе значення відрізняється від отриманого
        delay (10); // Чекаємо поки стан стабілізується - ігноруємо брязкіт
        current = digitalRead (pinButton); } // Прочитуємо стабілізовану значення
    return current; }
```

Після додавання функції, яка стабілізує стан кнопки, брязкіт ігноровано.

Апаратне усунення брязкоту.

Апаратне усунення брязкоту в переважній більшості випадків ефективніше програмного, з таких причин:

- апаратна реалізація завжди більш надійна,

- вивільнено ресурси процесора через відсутність необхідності додаткових обчислень,
- зменшено обсяг коду і спрощено процес налаштування.

Для апаратного усунення брязкоту реалізуємо коло, електрична принципова схема якого наведена на рис.2.9), що складено з підтягувального резистора R1 [39], конденсатора C2, додаткового резистора R2, тригера Шмітта та кнопки для комутації сигналу.

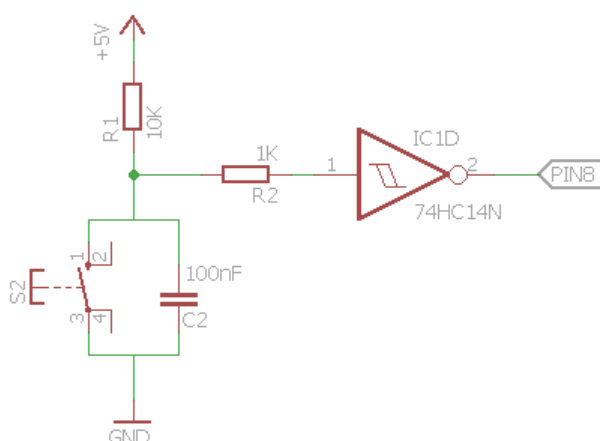


Рисунок 2.9 - Схема електрична принципова кола для усунення брязкоту

В разі замкнутого кола і заряду конденсатора, напруга U_c на його виводах зростає за формулою (1) від нуля до значення $U = 5$ В експоненціально (рис 2.10):

$$U_c = U(1 - e^{-\frac{t}{RC}}) \quad (1)$$

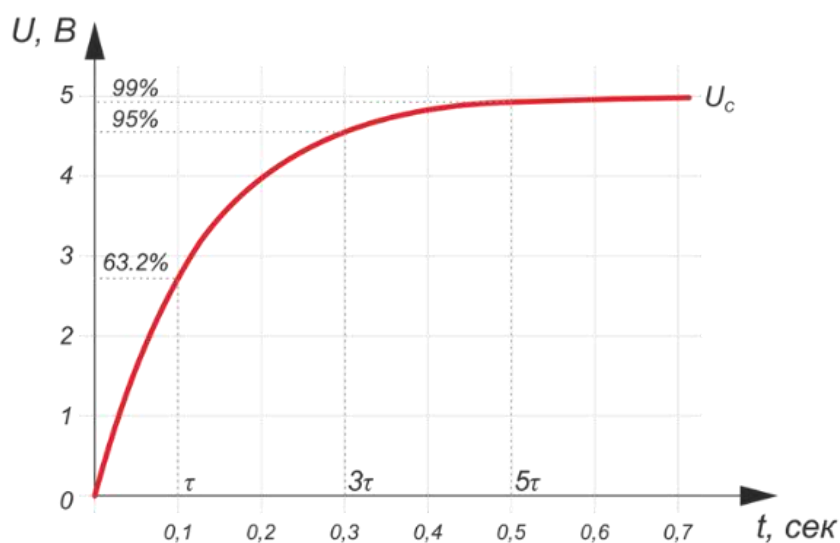


Рисунок 2.10 – Графік заряду конденсатора

У стані спокою на цифровому вході мікроконтролера є 5 В, а в разі натискання на кнопку, відбудеться розряд конденсатора. Після того, як кнопку відпущено, конденсатор почне заряджатися за час 5τ , де $\tau=RC$ [40].

Додатковий резистор R2 застосовано для стабілізації роботи кола - розряд конденсатора є повільнішим, а інвертувальний тригер Шмітта 74HC14N [41] перетворює аналогової сигнал у цифровий. Результат зафіксований осцилографом після усіх перетворень сигналу наведено на рис.2.11.



Рисунок 2.11 – Відсутність брязкоту у разі натискання кнопки, зафіксований осцилографом

2.3.2 Підключення декілька кнопок до одного аналогового входу

Наведено принцип збільшення цифрових портів, який застосовує лише один аналоговий порт. Схеми реалізовано за двома видами підключення: резистивна-послідовна та резистивна-паралельна. Принцип роботи схем засновано на читанні та інтерпретації аналого-цифровим перетворювачем мікроконтролера [42] індивідуальної напруги, формованої різними комбінаціями окремих ділянок електричного кола. Для комутації сигналу із входами мікроконтролера використано тактові кнопки з'єднання.

Схема №1 – резистивне-послідовне підключення

Принципово схема резистивно-послідовного підключення може бути виконано за двома варіантами (рис.2.12):

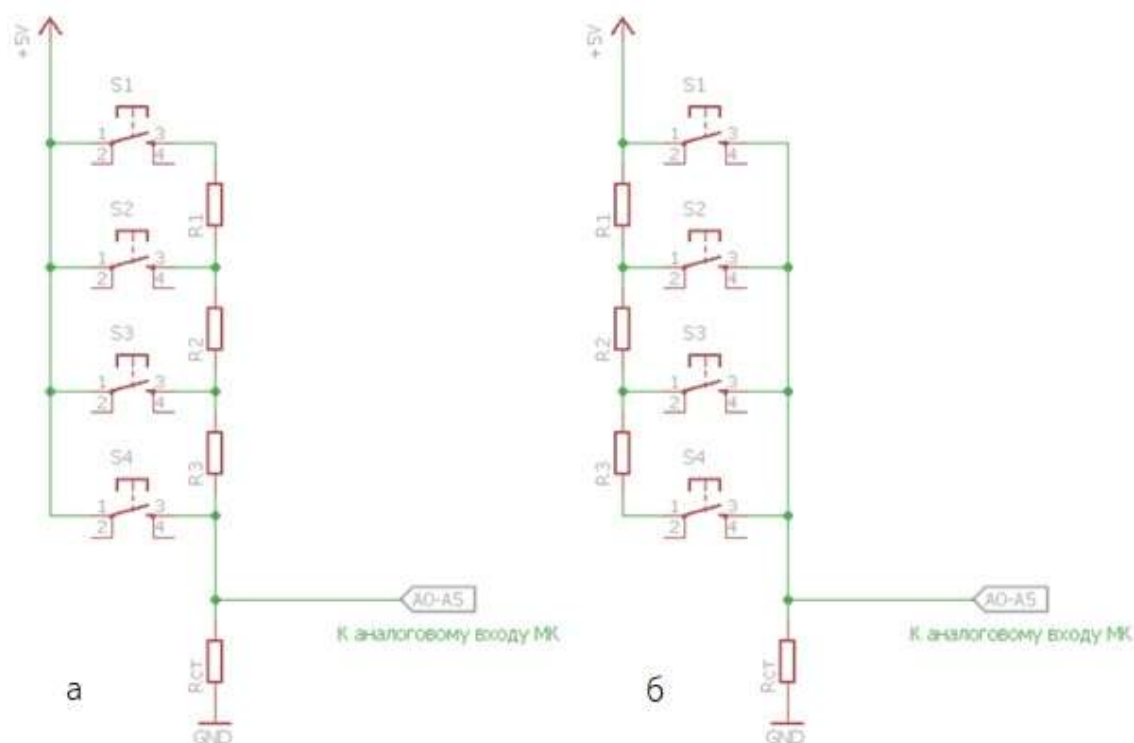


Рисунок 2.12 – Схеми електричні принципові комутації резисторів:

а – пряме підключення; б – зворотне підключення

Відмінність схем полягає в тому, що під час натискання кнопки S4 на схемі (рис.2.12а), аналоговий вхід сприймає максимальну напругу - 5 В, а кнопка S1 надає мінімальну напругу, знижену всіма резисторами в електричному колі. У схемі (рис.2.12б) максимальну напругу надає кнопка S1.

Наведені схеми (рис.2.12) - приклад послідовного з'єднання резисторів [43].

Загальний опір: $R_{\text{заг}} = R1 + R2 + \dots + Rn$

Під час натискання будь-якої кнопки у схемі отримуємо класичний резистивний дільник напруги [44] (рис.2.13):

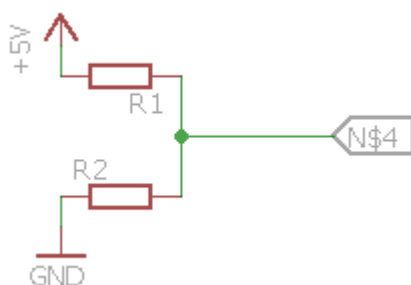


Рисунок 2.13 – Резистивний дільник напруги

Залежність напруги та опорів: $U_{\text{вих}} = U_{\text{вх}} \times R2 / (R1 + R2)$

З формули випливає, що залежність вихідної напруги $U_{\text{вих}}$ від опорів R_1 і R_2 не лінійна, а гіперболічна (рис 2.14). Для прикладу: $R_1 = 10 \text{ кОм}$, $R_2 = x$.

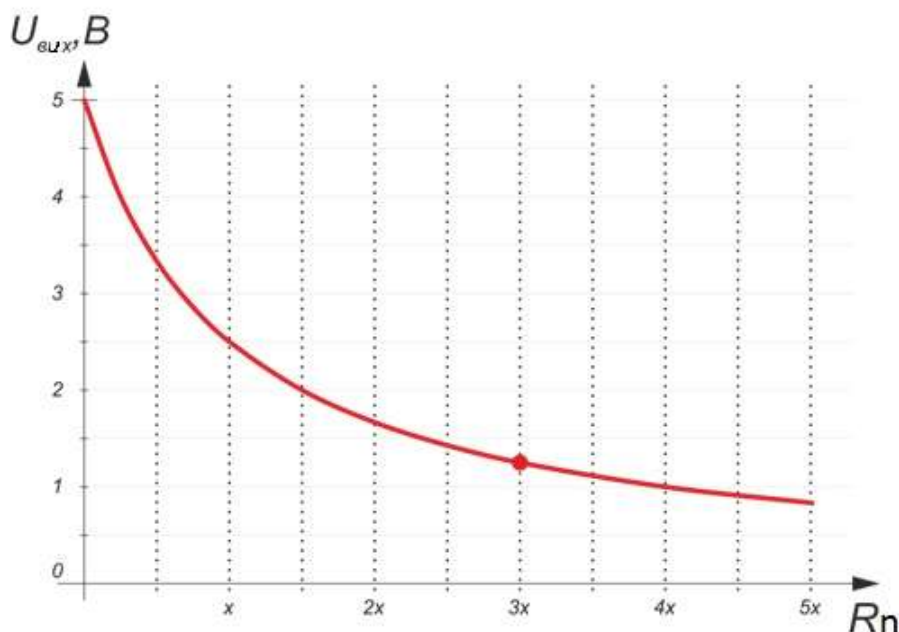


Рисунок 2.14 – Залежність напруги від опорів

Графік показує яким чином треба розрахувати значення опору резисторів залежно від кількості кнопок. Можливі 2 варіанти:

- підбирати опір, щоб сформувати інтервали з однакових проміжків напруги;
- використовувати резистори тільки одного значення опору.

У першому варіанті значення опору резисторів підібрано таким чином, щоб забезпечити однакові інтервали під час визначення значень напруги АЦП. Такий підхід є зручним для кількості кнопок 2^n , (за $n \geq 2$), тобто 4, 8, 16, 32, тощо. Зручність полягає в тому, що дані роботи АЦП отримані в результаті бітового зсуву [45] можна округлити, при цьому максимально нехтувати похибкою АЦП і значеннями опорів резисторів.

Використовувати цей варіант можна і з кількістю кнопок не кратним 2^n . В цьому режимі похибка АЦП усунена програмно - впровадженням значення помилки.

Недолік такого способу полягає в обмеженні кількості кнопок - за збільшення їх числа, опір резисторів R_n , що формують низькі напруги (менше 1 В), зростають в геометричній прогресії (рис.2.15).

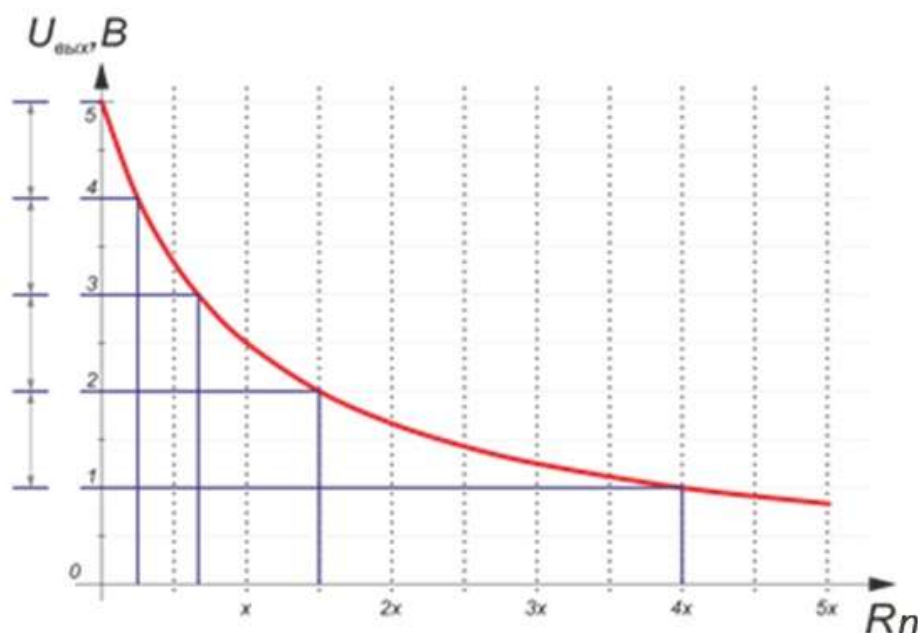


Рисунок 2.15 - Залежність з однаковими інтервалами для напруги

Формула для розрахунку номіналів резисторів для установки однакових проміжків напруги для кількості кнопок k , і пов'язаних з номіналом стягувального резистора [46] $R_{ст}$ (якщо прийняти номінал $R_{ст}$ за одиницю, то отримані коефіцієнти можна використовувати під час розрахунку за іншому заданому опорі

$$R_n = \frac{U_{вх} \times R_{ст}}{u \times (k - n)} - R_{ст} - \sum_{i=1}^{n-1} R_i, \text{ при } n \in \mathbb{Z} \mid 1 \leq n \leq k - 1$$

стягувального резистора):

$$u = \frac{U_{вх}}{k} \text{ — крок дискретності вихідної напруги}$$

Оскільки першу кнопку застосовано без резистора, то резистор нульового опору опускаємо, звідси $n \leq k-1$ - ця комутація дасть результат рівний $U_{вх} = 5\text{В}$.

Розглянемо розрахунок номіналів на прикладі 5 кнопок і 4 резисторів, який стягує резистор $R_{ст} = 10 \text{ кОм}$:

$$u = \frac{U_{ex}}{k} = \frac{5\text{В}}{5} = 1\text{В}$$

$$R_1 = \frac{5\text{В} \times 10\text{кОм}}{1\text{В} \times (5 - 1)} - 10\text{кОм} - 0\text{кОм} = 2,5\text{кОм}$$

$$R_2 = \frac{5\text{В} \times 10\text{кОм}}{1\text{В} \times (5 - 2)} - 10\text{кОм} - (2,5\text{кОм}) = 4,167\text{кОм}$$

$$R_3 = \frac{5\text{В} \times 10\text{кОм}}{1\text{В} \times (5 - 3)} - 10\text{кОм} - (2,5\text{кОм} + 4,167\text{кОм}) = 8,333\text{кОм}$$

$$R_4 = \frac{5\text{В} \times 10\text{кОм}}{1\text{В} \times (5 - 4)} - 10\text{кОм} - (2,5\text{кОм} + 4,167\text{кОм} + 8,333\text{кОм}) = 25\text{кОм}$$

Таким чином, для цієї схеми потрібні резистори приблизних опорів - 2,5 кОм; 4,167 кОм; 8,333 кОм; 25 кОм. Похибка для значень ні є критичною.

Інший спосіб це встановлення резисторів однакових значень опору. Зауважимо, що цей опір буде менше опору стягувального резистора x . Тоді перші кнопки є в діапазоні, де відмінності між напругами максимальні. Діапазон напруги визначаємо за формулою, за значенням номіналу стягувального резистора $R_{ст}$ і в інших резисторів R :

$$U_{\text{вых}_n} = U_{\text{ex}} \times \frac{R_{\text{ст}}}{R \times n + R_{\text{ст}}}$$

В разі такого підключення дискретність напруги знижується. Для запобігання цього можна за умови, що різниця між напругою є критично мінімальною, почати встановлювати резистори більшого значення, за збільшенням дискретності (рис.2.16):

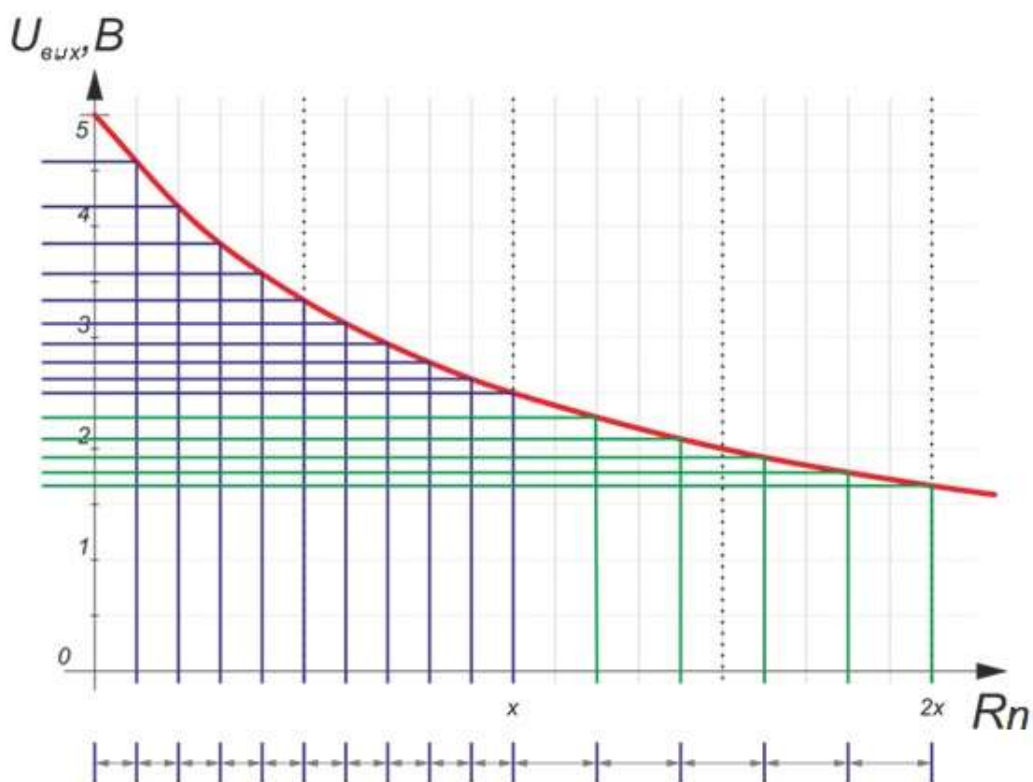


Рисунок 2.16 - Залежність з однаковими номіналами резисторів

Основною відмінною рисою резистивно-послідовної схеми є ігнорування натискання кількох кнопок (рис. 2.17). Результат завжди буде вказувати на натискання єдиної кнопки з меншим опором по відношенню до сумарного опору інших натиснутих кнопок:

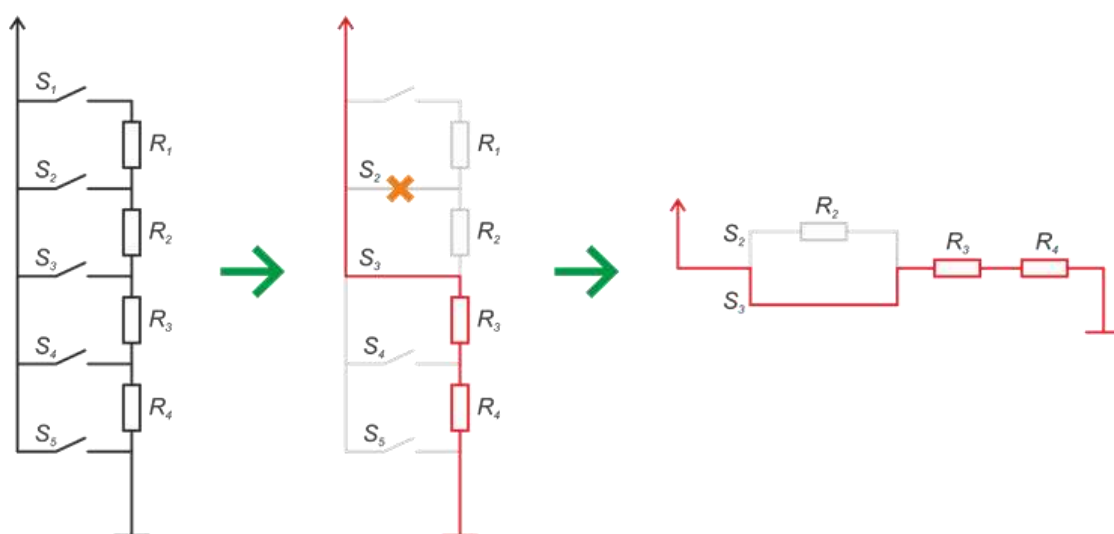


Рисунок 2.17 – Приклад реалізації комутації сигналу

Переваги резистивно-послідовного підключення схеми:

- передбачувана поведінка;
- порівняна легкість розрахунків;
- натискання кількох кнопок одночасно призводить до передбачуваному результату;
- можливі два варіанти реалізації: із однаковим кроком зміни вихідної напруги або із однаковими значеннями опорів використовуваних резисторів.

Недоліки резистивно-послідовного підключення схеми:

- накопичується похибка підібраних значень опорів резисторів;
- складність розрахунків для рівномірної дискретності значень напруги;
- обмеження кількості комутації в діапазоні напруг до 1 В;
- принциповий порядок розташування резисторів на схемі.

Схема №2 - резистивне-паралельне підключення

Особливість цього підключення полягає в тому, що резистор кожної кнопки задає необхідне значення напруги на виході. Опір резисторів розраховують як для резистивно-послідовної схеми підключення, де резистори мають різні значення. Під час натискання двох або кількох кнопок, генеровано індивідуальну напругу, яку визначає паралельне підключення [47] резисторів:

$$\frac{1}{R_{\text{общ}}} = \frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n}$$

або

$$R_{\text{общ}} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n}}$$

Розпізнавання одночасно натиснутих кнопок є перевагою за малої їх кількості, і є практично непередбачуваним за великій кількості комутацій. Розрахунок значень опорів схожий з розрахунком резистивно-послідовної схеми, за винятком, що кожне наступне значення опору не потрібно коригувати стосовно отриманих вище значень:

$$R_n = \frac{U_{\text{вх}} \times R_{\text{ст}}}{U_{\text{ввих}_n}} - R_{\text{ст}}$$

Наприклад, схема з трьома комутаціями має генерувати 1 В; 2,5 В і 4 В. Номінал стягувального резистора 10 КОм, вхідна напруга 5 В. Розрахунок опорів резисторів виконуємо таким чином:

$$R_1 = \frac{5\text{В} \times 10_{\text{КОм}}}{1\text{В}} - 10_{\text{КОм}} = 40_{\text{КОм}}$$

$$R_2 = \frac{5\text{В} \times 10_{\text{КОм}}}{2,5\text{В}} - 10_{\text{КОм}} = 10_{\text{КОм}}$$

$$R_3 = \frac{5\text{В} \times 10_{\text{КОм}}}{4\text{В}} - 10_{\text{КОм}} = 2,5_{\text{КОм}}$$

Особливість цієї схеми полягає в тому що, під час натискання декілька кнопок водночас, АЦП мікроконтролера приймає інше значення напруги і як наслідок збільшує кількість можливих комутацій для тієї ж кількості кнопок. Для схеми з трьома кнопками цих комбінацій 4: (1+2, 1+3, 2+3, 1+2+3).

Кількість комбінацій натиснень кнопок визначено за формулою числа сполучень [48] із загального числа n -об'єктів по k -штук в комбінації:

$$C_n^k = \frac{n!}{(n-k)! \times k!}$$

Оскільки виключені натискання по одній кнопці (принцип дії за умовчанням), і можливі натискання не тільки 2, але і всіх кнопок відразу. Загальна формула для знаходження числа сполучень знаходиться шляхом складання числа сполучень 2 кнопок, числа поєднань 3 кнопок, тощо:

$$C_{n_{\text{обш}}} = C_n^2 + C_n^3 + \dots + C_n^n$$

$$C_{n_{\text{обш}}} = \sum_{k=2}^n \frac{n!}{(n-k)! \times k!}$$

Наприклад, для 5 комутацій, окрім 5 відокремлених натискань по одній кнопці, загальна кількість комбінацій розраховано таким чином. Кількість комбінацій натиснень по дві кнопки (1+2, 1+3, 1+4, 1+5, 2+3, тощо):

$$C_5^2 = \frac{5!}{(5-2)! \times 2!} = 10$$

Кількість комбінацій натиснень 3 кнопок (1+2+3, 1+2+4, 1+2+5, 2+3+4, тощо):

$$C_5^3 = \frac{5!}{(5-3)! \times 3!} = 10$$

Кількість комбінацій натиснень 4 кнопок (1+2+3+4, 1+2+3+5, тощо):

$$C_5^4 = \frac{5!}{(5-4)! \times 4!} = 5$$

Також кількість одночасних натискань 5 кнопок - 1. Разом:

$$C_{n_{\text{общ}}} = 10 + 10 + 5 + 1 = 26$$

Таким чином маємо 26 комбінацій, окрім 5 стандартних. Для 6 комутацій можливих комбінацій є 57, без урахування 6 основних.

Переваги резистивно-паралельного підключення схеми:

- простота підбору значень опорів;
- не важливий порядок установки резисторів;
- можливо нарощування додаткового функціоналу за рахунок інтерпретацій комбінацій комутацій.

Недоліки резистивно-паралельного підключення схеми:

- при включенні в схему більше 5 кнопок, поведінка схеми при одночасному натисканні двох і більше кнопок стає непередбачувана.

2.3.3 Збільшення аналогових входів завдяки мультиплексору CD4051

CD4051 - це 8 каналний аналоговий CMOS мультиплексор / демультимплексор. Дана мікросхема допоможе вирішити проблему, коли не

вистачає аналогових входів. Використовуючи 1 аналоговий і 3 цифрових входу можна отримати 8 аналогових входів.

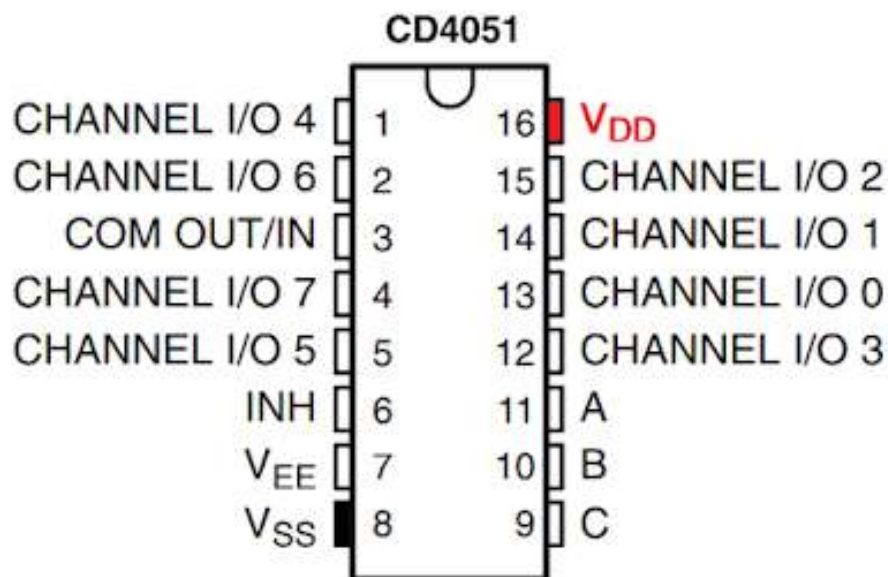


Рисунок 2.18 – Мультиплексор CD4051 [49]

Мультиплексором – це пристрій, який дозволяє передавати сигнал з декількох входів на єдиний вихід [49]. Демультиплексор, навпаки, передає сигнал з єдиного входу на один з інформаційних виходів. CD4051 поєднує в собі ці дві функції. Принцип роботи наведено на рисунку 2.19:

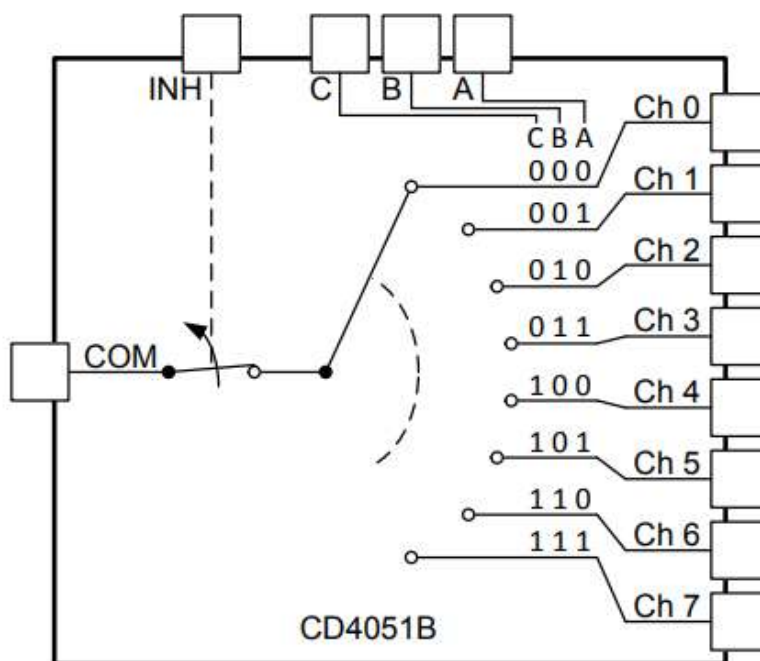


Рисунок 2.19 - Функціональна схема CD4051 [49]

Висновок COM підключається до аналогового виводу мікроконтролера. Значення на цифрових входах А, В, С визначають який з каналів буде з'єднаний з виводом COM. З'єднання можливо тільки при наявності дозволяючого сигналу (низький рівень) на вході INH.

За допомогою мікросхеми можна, наприклад, керувати 8 світлодіодами або зчитувати значення 8 потенціометрів. Можна збільшити кількість п'єзоелементів, що підключаються до Ардуїно для створення MIDI барабанної установки. Для прикладу підключено 8 фоторезисторів до Ардуїно. Схема підключення 8 фоторезисторів до Ардуїно з використанням аналогового комутатора CD4051 наведено на рисунку 2.20.

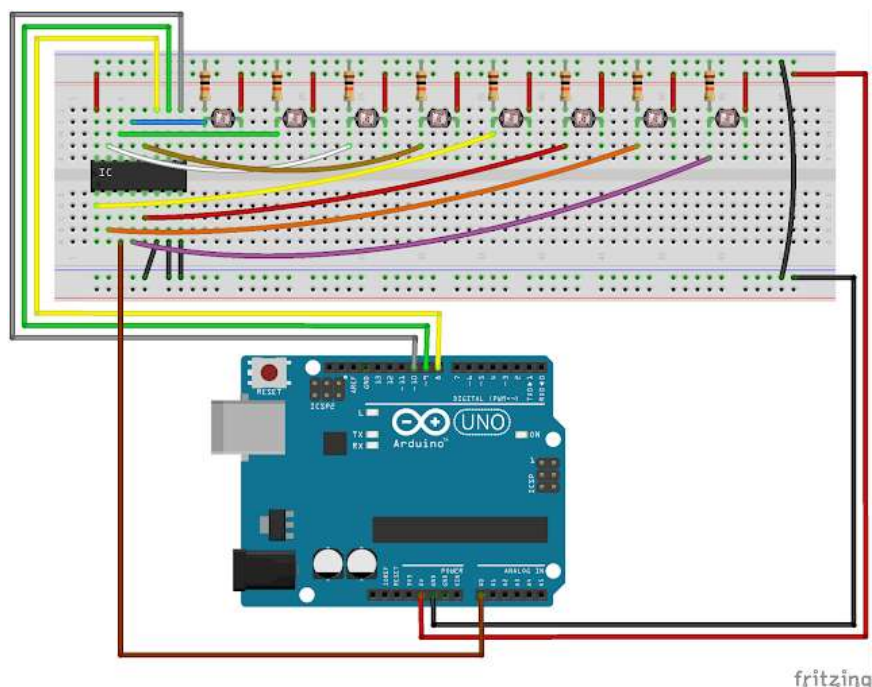


Рисунок 2.20 - Схема підключення 8 фоторезисторів до Ардуїно

Вивід INH відповідає за дозвіл зчитування. Якщо під'єднати його до землі, то зчитування / запис сигналу через CD4051 будуть завжди дозволені. Вивід COM підключений до піну A0 Ардуїно, виводи А, В і С до цифрових виводів 8, 9 і 10 відповідно. Щоб не встановлювати значення кожного з цих пинов функцією digitalWrite можна звертатися безпосередньо до регістру В, який відповідає за значення виводів з 8 по 13.

// Лістинг програми для роботи із мультиплексором CD4051

byte AnalogIn = A0; // CD4051 pin 3 (Common in/out)

```

void setup() {
  Serial.begin(9600);
  pinMode(8, OUTPUT); // CD4051 pin 11 (A)
  pinMode(9, OUTPUT); // CD4051 pin 10 (B)
  pinMode(10, OUTPUT); // CD4051 pin 9 (C)
}
void loop() {
  for (byte i = 0; i < 8; i++) {
    PORTB = (PORTB & B11111000) ^ i;
    Serial.print(i);
    Serial.print(": ");
    Serial.println(analogRead(AnalogIn));
  }
  Serial.println();
  delay(1000); }

```

Після прошивки даного скетчу в Ардуіно, в моніторі порту будуть відображатися значення, прийняті з фоторезисторів.

Таким чином за допомогою CD4051 можна збільшити кількість аналогових висновків в Ардуіно. А також можна підключити декілька мікросхем CD4051. Для цього виводи Ардуіно, що відповідають за вибір каналу, підключаються до виводів А, В і С всіх мікросхем, а вибір активної мікросхеми здійснюється установкою дозволяючого сигналу на вході INH, наприклад, за допомогою дешифратора.

2.3.4 Збільшення цифрових виходів завдяки зсувному регістру 74НС595

В моєму проекті потрібно більше виходів, ніж може запропонувати контролер Arduino. Для вирішення цієї проблеми, використано зсувний регістр [50]. В даному прикладі використано 74НС595.

74НС595 - восьмирозрядний зсувний регістр з послідовним введенням, послідовним або паралельним виведенням інформації, з тригером-клямкою і трьома станами на виході.

Цей регістр дає змогу контролювати 8 виходів, використовуючи всього кілька виходів на самому контролері. При цьому кілька таких регістрів можна об'єднувати послідовно для каскадування.

В даній схемі використано принцип синхронізованої послідовної передачі сигналів. Необхідні значення сигналу (біти HIGH або LOW) передаються в регістр один за іншим, при цьому регістр отримує синхронізований сигнал, який змушує його зчитувати сигнал з входу. Коли байт (1 байт = 8 біт) зчитано, значення всіх 8 біт розподілені по виходах. Тобто передаємо в регістр сигнали послідовно, на виходах регістра маємо паралельно 8 сигналів.

74НС595 може віддавати сигнали не тільки паралельно, а й послідовно. Це необхідно при об'єднанні кількох регістрів, для отримання 16 і більше виходів. У цьому випадку перші 8 біт сигналу передаються на наступний регістр для паралельного виводу на ньому.

Три можливих стану на виході, згадані вище, означають, що вихід регістра може мати не тільки логічний нуль або одиницю (HIGH або LOW), але і бути в високоомному (високоімпедансному) стані - коли вихід відключений від схеми.

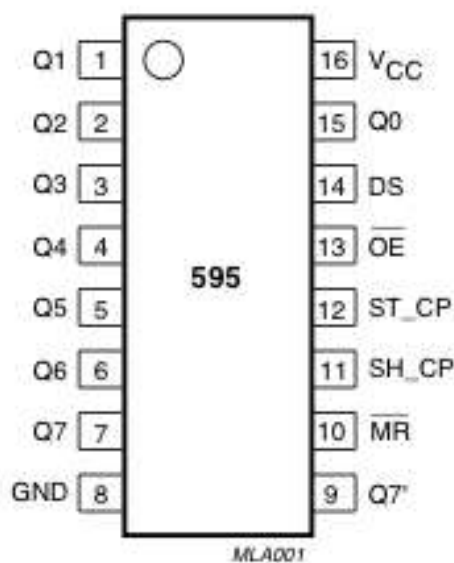


Рисунок 2.21 - Зсувний регістр 74НС595

Входи / виходи регістра

- Вивід 1-7, 15: Q0 - Q7 Паралельні виходи
- Вивід 8: GND Земля
- Вивід 9: Q7 Вихід для послідовного з'єднання регістрів
- Вивід 10: MR Скидання значень регістра.
- Вивід 11: SH_CP Вхід для тактових імпульсів

- Вивід 12: ST_CP Сінхронізація виходів
- Вивід 13: OE Вхід для перемикавання стану виходів з високоомного в робочий стан
- Вивід 14: DS Вхід для послідовних даних
- Вивід 16: Vcc Живлення

Схема підключення з одним регістром наведено на рисунку 2.22

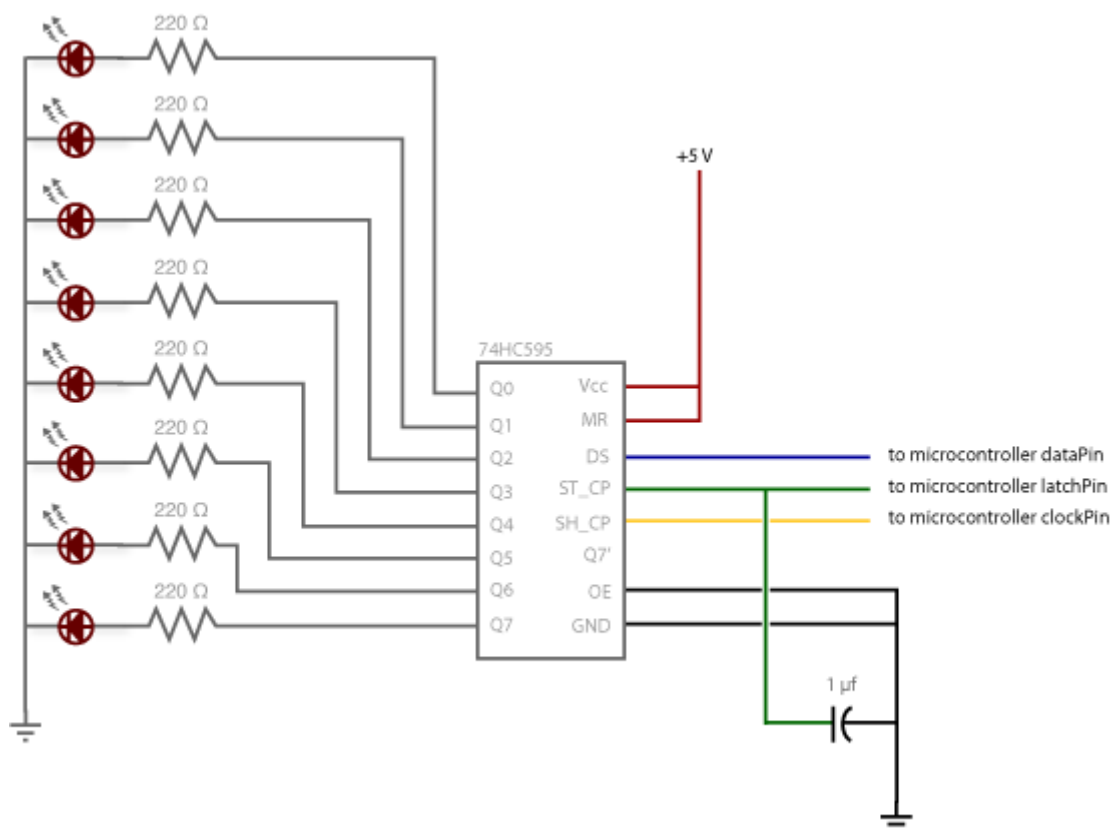


Рисунок 2.22 – Підключення регістру до Арудіно

- GND (вивід 8) на землю
- Vcc (вивід 16) до живлення 5V
- OE (вивід 13) на землю
- MR (вивід 10) до живлення 5V
- DS (вивід 14) з 11-м цифровий вихід Arduino (на схемі синій провід)
- SH_CP (вивід 11) з 12-им цифровим виходом (жовтий провід)
- ST_CP (вивід 12) с 8-им (зелений провід)

При використанні регістрів відмінних від 74HC595 слід звіритися з документацією і перевірити схему підключення.

Лістинг програми для роботи із зсувний регістром

```
// Пін підключений до ST_CP входу 74HC595
int latchPin = 8;
// Пін підключений до SH_CP входу 74HC595
int clockPin = 12;
// Пін підключений до DS входу 74HC595
int dataPin = 11;
void setup () {
    // встановлюємо режим OUTPUT
    pinMode (latchPin, OUTPUT);
    pinMode (clockPin, OUTPUT);
    pinMode (dataPin, OUTPUT);
}
void loop () {
    // відраховуємо від 0 до 255 і відображаємо значення на світлодіоді
    for (int numberToDisplay = 0; numberToDisplay <256; numberToDisplay ++ )
    {
        // встановлюємо синхронізацію "засувки" на LOW
        digitalWrite (latchPin, LOW);
        // передаємо послідовно на dataPin
        shiftOut (dataPin, clockPin, MSBFIRST, numberToDisplay);
        // закриваємо регістр, тим самим встановлюючи значення на виходах
        digitalWrite (latchPin, HIGH);
        // пауза перед наступною ітерацією
        delay (500); } }
```

Тимчасова діаграма сигналів регістру та таблиця логіки наведено на рисунках 2.23 та 2.24 відповідно. Коли clockPin перемикається з LOW на HIGH, регістр зчитує значення з DS виводу. Зчитуванні дані записуються у внутрішню пам'ять. Коли latchPin перемикається з LOW на HIGH, дані "замикаються", тобто передаються на виходи регістра, включаючи світлодіоди.

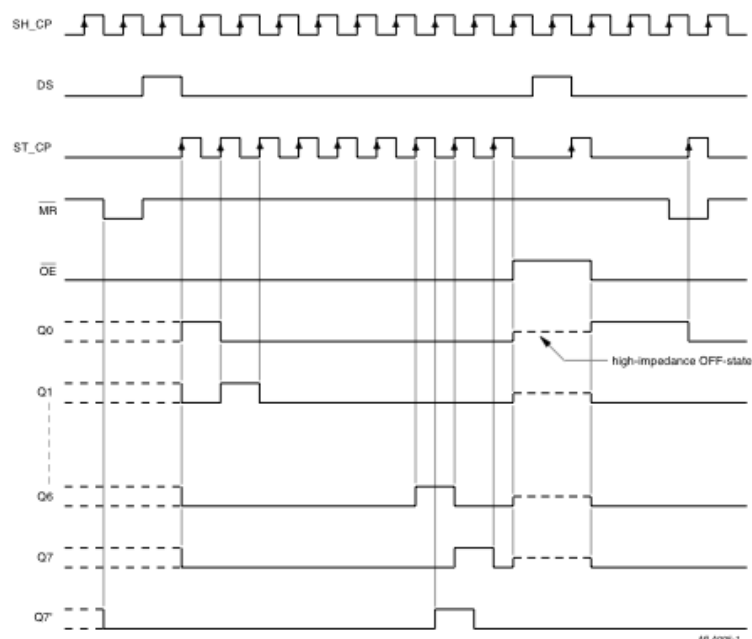


Рисунок 2.23 - Тимчасова діаграма сигналів регістру

FUNCTION TABLE

See note 1.

INPUT					OUTPUT		FUNCTION
SH_CP	ST_CP	OE	MR	DS	Q7'	Qn	
X	X	L	L	X	L	n.c.	a LOW level on MR only affects the shift registers
X	↑	L	L	X	L	L	empty shift register loaded into storage register
X	X	H	L	X	L	Z	shift register clear; parallel outputs in high-impedance OFF-state
↑	X	L	H	H	Q6'	n.c.	logic high level shifted into shift register stage 0; contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal Q6') appears on the serial output (Q7')
X	↑	L	H	X	n.c.	Qn'	contents of shift register stages (internal Qn') are transferred to the storage register and parallel output stages
↑	↑	L	H	X	Q6'	Qn'	contents of shift register shifted through; previous contents of the shift register is transferred to the storage register and the parallel output stages

Note

1. H = HIGH voltage level;
 L = LOW voltage level;
 ↑ = LOW-to-HIGH transition;
 ↓ = HIGH-to-LOW transition;
 Z = high-impedance OFF-state;
 n.c. = no change;
 X = don't care.

Рисунок 2.24 – Таблиця логіки регістру [50]

Приклад використання каскаду зсувних регістрів

Регістри можна об'єднувати у каскади, доводячи кількість виходів до 16. Для цього потрібно підключити другий регістр до живлення і загальної землі так само, як і перший. DS вхід (вивід 14) підключається до Q7' виходу (вивід 9) першого регістра (синій провід). А SH_CP (вивід 11) і ST_CP (вивід 12)

3. РЕАЛІЗАЦІЯ ПРИСТРОЮ ДЛЯ КОНТРОЛЬОВАНОГО НАГРІВАННЯ ВОДИ, ЯК ЕЛЕМЕНТ ІНТЕРНЕТУ РЕЧЕЙ

3.1. Конструювання пристрою контрольованого нагрівання води

Конструювання пристрою починають з визначення усіх необхідних периферійних пристроїв, які застосовані для реалізації системи контрольованого нагрівання води, а саме:

- потенціометр;
- фоторезистор;
- датчик температури DS18B20;
- датчик тиску MPXV5050;
- інфрачервоний модуль;
- твердотільне реле;
- драйвер двигуна;
- сервопривід;
- водяна помпа;
- RGB – світлодіод;
- модуль звуку;
- дисплей LCD 2004;
- модуль Bluetooth;
- тактові кнопки;
- перемикачі;
- проводи для з'єднання.

А також потрібен мікроконтролер Arduino Uno, щоб керувати усіма периферійними пристроями. Усі вище перелічені компоненти зображені на рисунку 3.1.



Рисунок 3.1 – Компоненти для конструювання пристрою

3.1.1. Схема електрична принципова пристрою

Периферійні пристрої підключаються до мікроконтролера за принциповою схемою, яка зображена на рисунку 3.2.



Рисунок 3.2 - Схема електрична принципова

3.1.2. Підключення електричних компонентів

Перш ніж спаювати між собою електричні компоненти, потрібно під'єднати їх між собою на макетній платі для того щоб впевнитися в правильній роботі пристрою. Приклад попереднього з'єднання тактових кнопок та потенціометрів до мікроконтролера на макетній платі зображено на рисунку 3.3

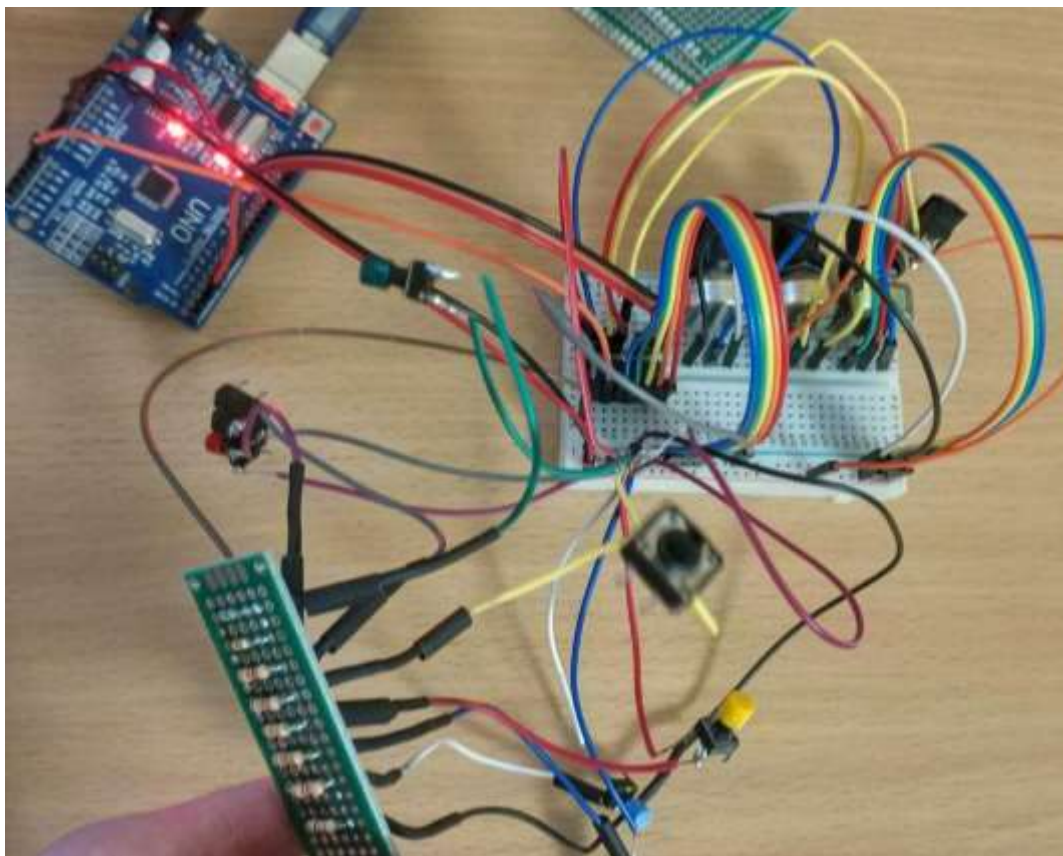


Рисунок 4.3 – З'єднання компонентів на макетній платі

Після перевірки працездатності попереднього з'єднання електричних компонентів, спаяємо усі інші периферійні пристрої між собою. Обов'язковою умовою є об'єднання землі кожного компонента в загальний провід. Земля забезпечує синхронізацію периферійних пристроїв між собою та забезпечує коректну роботу системи. З'єднання усіх компонентів наведено на рисунку 3.4.

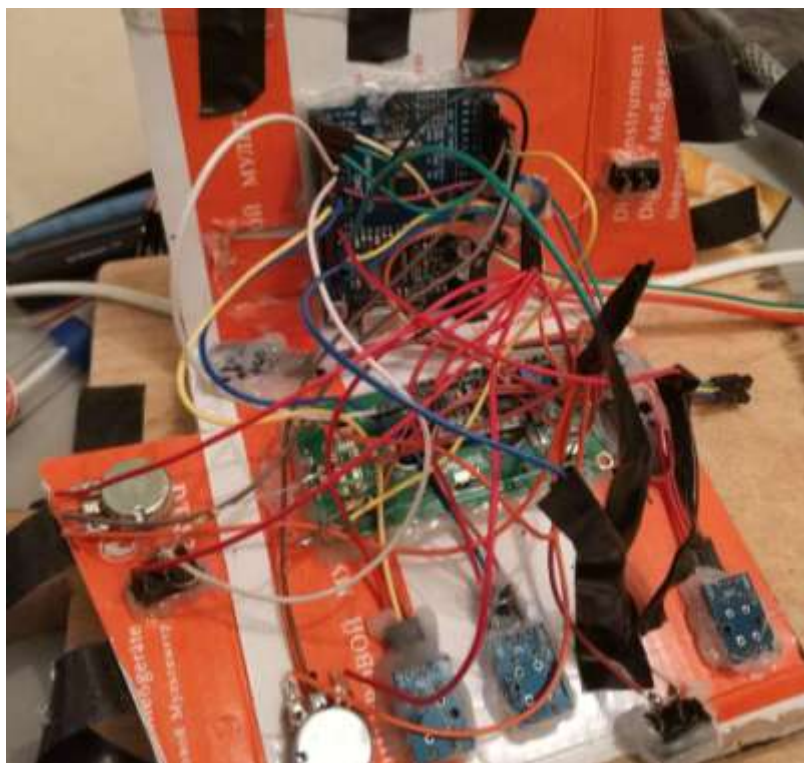


Рисунок 3.4 – З'єднання усіх електричних компонентів між собою

3.1.3. Макет для корпусу пристрою

Макет корпусу для системи контролюваного нагрівання води створено із ПВХ паперу та розфарбовано у чорний колір. Усі частини макету з'єднанні термопластичним клеєм. Макет має параметри:

- ширина – 127 мм;
- довжина – 105 мм;
- висота – 35 мм;
- вага – 192 г.

На зовнішній частині макеті розташовані елементи:

- тактові кнопки;
- потенціометри;
- перемикачі;
- фоторезистор;
- модуль звуку;

- вивід для підключення живлення;
- вивід для завантаження прошивки;
- RGB світлодіод.

Зовнішній вигляд макета пристрою наведено на рисунку 3.5



Рисунок 3.5 – Зовнішній вигляд макета пристрою

3.1.4. Приклад програмної реалізації

Для функціонування пристрою розроблено код програми на мові C++, який завантажений на мікроконтролер Arduino Uno через компілятор Arduino IDE, розроблений тим самим виробником – Arduino.cc.

Розроблений код програми наведено у Додатку А.

Перевірка коду на працездатність у компіляторі Arduino IDE:

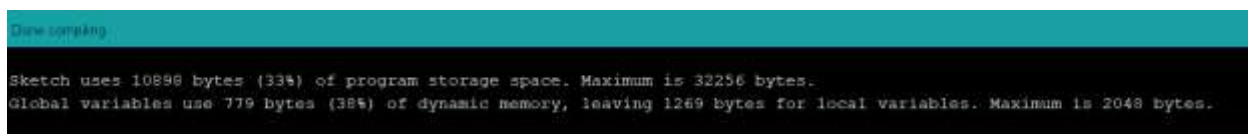


Рисунок 3.6 – Консоль компілятора Arduino IDE

3.1.5. Принцип дії пристрою

Реалізований пристрій має можливість вимірювати у реальному часі рівень води із кроком у 100 мл та температуру із кроком у 1°C. Тому перше за все реалізовано система безпеки, яка забороняє працювати пристрою коли до системи не під'єднаний чайник або в чайнику води менш, ніж мінімальний рівень. Використовуючи можливості вимірювання рівня та температури, в пристрої реалізовано такі режими роботи:

1.) Нагрівання води до 100°C

Під час натискання тактової кнопки «Start» мікроконтролер подає команду до реле замкнутися та датчик температури вимірює стан нагрівання води. Коли температура досягає 100°C , реле автоматично відмикається, тим самим зупиняє процес нагрівання води.

2.) Нагрівання води до встановленої температури.

Для встановлення температури потрібно ручкою потенціометра встановити значення тієї температури, до якої потрібно нагріти воду. Далі при натисканні тактової кнопки «Set Temp» пристрій виконує усі операції, як і в першому режимі, але процес нагрівання зупиниться, коли датчик температури зафіксує встановлене значення на потенціометрі. Під час цього режиму, значення потенціометра можна корегувати і таким чином змінювати встановлену температуру.

3.) Утримання температури води на встановленому значенні.

Для утримання температури на встановленому значенні, потрібно використовувати тій самий потенціометр, що і у другому режимі. Після встановлення значення температури, натискаємо тактову кнопку «Hold Temp» для того щоб запустити режим утримання температури. Під час досягнення встановленої температури, процес нагрівання зупиняється до того часу, коли температура води не стане меншою, ніж встановлена на потенціометру. Вимкнути цей режим можна кнопкою «Stop» або включенням іншого режиму.

4.) Встановлення рівня води до заданого значення.

Іншим потенціометром, який відповідає за значення рівня води встановлюємо необхідний об'єм води, який має міститися в пристрої нагрівання води. Тактовою кнопкою «Set Water» запускаємо процес набору води. Цей процес зупиняється, коли рівень води досягає встановленого значення. Встановлення води можливе лише, коли пристрій присутній в системі та не нагріває воду.

5.) Утримання рівня води на встановленому значенні.

Для утримання рівня води, треба встановити значення рівня на тому ж потенціометрі, який використано в четвертому режимі. Після натискання тактової кнопки «Hold Water» пристрій перевіряє наявність пристрою в системі та перевіряє відсутність нагрівання води в даний момент часу. Якщо ці умови виконані та рівень води в пристрої менший за встановлений, водяна помпа підкачує воду до заданого значення.

6.) Режим екстреної зупинки

При натисканні тактової кнопки «Stop» зупиняється будь-який із вище описаних режимів. Цей режим використовується як для вимкнення режимів утримання температури та рівня води, так і для екстреної зупинки нагрівання води чи набору води до пристрою.

Інші технічні властивості пристрою:

- Є можливість вибору автоматичного чи ручного керування яскравістю LCD дисплею. При виборі ручного режиму – можна змінювати яскравість окремим потенціометром. При виборі автоматичного – яскравість змінюється відповідно до освітленості приміщення, де встановлена система.
- Модуль звука створює повідомлення про кінець нагрівання води та встановлення відповідного рівня води в системі, а також має деякі інші системні сигнали.
- RGB-світлодіод інформує про поточний режим системи та змінює свій колір від холодного синього до теплого червоного, відповідно до температури води під час нагрівання.

3.2. Розроблення програмного середовища на смартфон для керування пристроєм

Системою контрольованого нагрівання води можна керувати через мобільний додаток на смартфоні із операційної системою Android [51]. Для створення мобільного додатку використано середовище розробки App Inventor.

3.2.1. Середовище розробки MIT App Inventor 2

MIT App Inventor - це інтуїтивно зрозуміле середовище візуального програмування, яке дозволяє створювати повністю функціональні додатки для смартфонів та планшетів [52]. App Inventor 2 на основі блоків сприяє створенню складних додатків за значно менший час, ніж традиційні середовища програмування. Проект MIT App Inventor спрямований на демократизацію розробки програмного забезпечення шляхом надання можливості всім людям, переходити від споживання технології до створення технологій.

Для програмування в App Inventor використовується графічний інтерфейс користувача, візуальна мова програмування дуже схожа на мову Scratch і StarLogo TNG. Компілятор, що трансліює візуальну блокову мову App Inventor в байт-код Android, заснований на фреймворку GNU для реалізації динамічних мов Kawa, що реалізує Scheme для платформи Java (Android). Даний програмний продукт розроблено в компанії Google з використанням Java-бібліотеки Open Blocks, розробленої в MIT

Невелика команда співробітників та студентів CSAIL під керівництвом професора Хела Абелсона підтримує це безкоштовне онлайн-середовище розробки додатків, яким користуються понад 6 мільйонів зареєстрованих користувачів.

3.2.2. Приклад програмної реалізації

Спочатку створено графічний дизайн мобільного додатку в розділі Designer. Інтерфейс середовища App Inventor 2 зображено на рисунку 3.7.

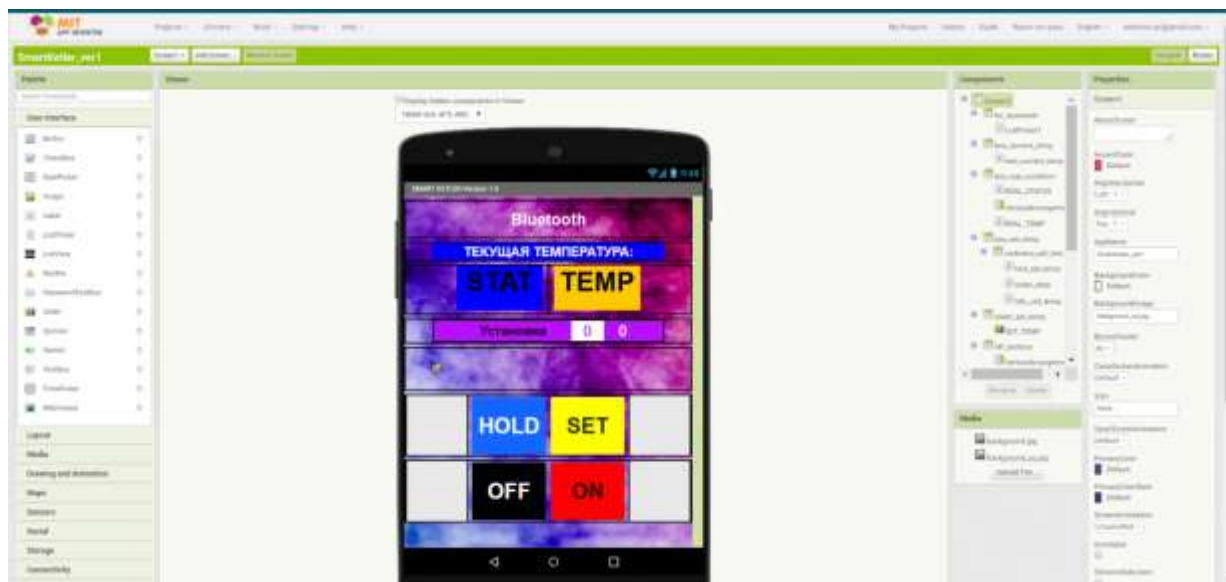


Рисунок 3.7 – Середовище розробки App Inventor 2

Програмний код створюється в розділі Blocks, який використовує блочні схеми для програмування окремих елементів графічного дизайну. Приклад блочного коду для додатку, яким можна керувати розроблену систему контролюваного нагрівання води зображено на рисунку 3.8.

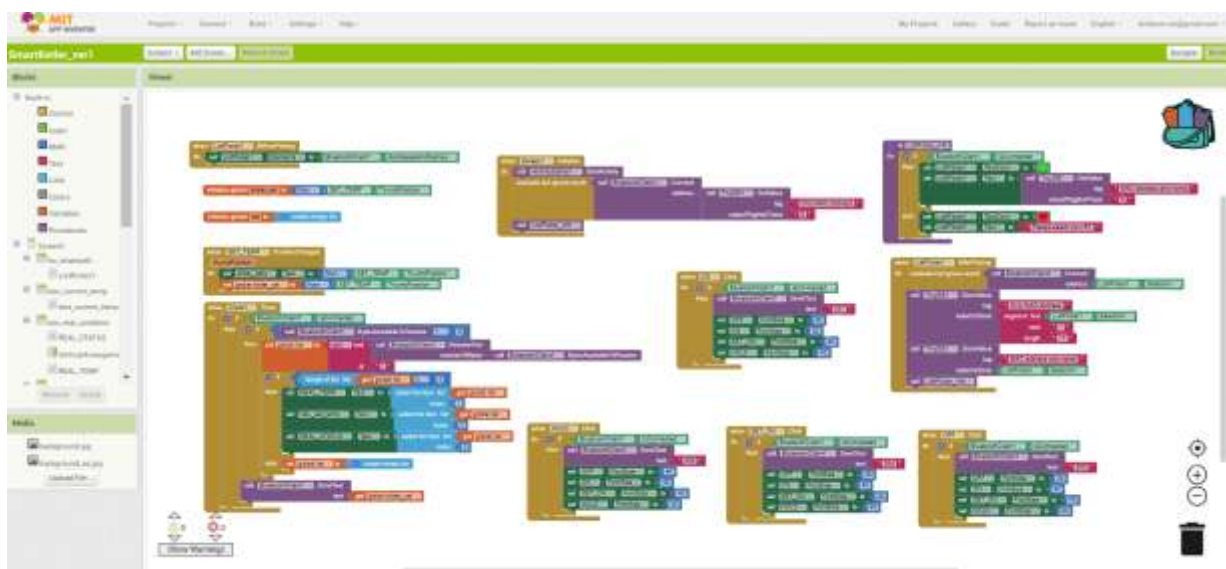


Рисунок 3.8 – Програмний код мобільного додатку із блоків

Для перевірки працездатності додатку створюємо .apk файл, який завантажуюмо на смартфон та під'єднуємось до Bluetooth модуля розробленого пристрою системи нагрівання води. Захоплення екрану зі смартфона із підтвердженням працездатності мобільного додатку зображено на рисунку 3.9.



Рисунок 3.9 – Мобільний додаток для керування пристроєм

3.2.3. Принцип дії програми

Принцип дії мобільного додатку повністю дублює керування пристроєм тактовими кнопками. Весь обмін інформацією здійснюється через Bluetooth протокол. Під час відкриття мобільного додатку, він автоматично розпочинає

пошук заздалегідь встановленого IP-адресу пристрою та під'єднується автоматично. Повзунки встановлення значення температури та рівня води відповідають потенціометрам на пристрої. Кнопки у додатку запускають відповідні режими роботи пристрою, а саме:

- «Start» - Нагрівання води до 100°C;
- «Set Temp» - Нагрівання води до встановленої температури;
- «Hold Temp» - Утримання температури води на встановленому значенні;
- «Set Water» - Встановлення рівня води до заданого значення;
- «Hold Water» - Утримання рівня води на встановленому значенні;
- «Stop» - Режим екстреної зупинки.

Висновки до розділу:

У цьому розділі запропонована удосконалена система для контрольованого нагрівання води. Система заснована на мікроконтролері Arduino Uno та периферійних пристроях, які використовуються для вимірювання значення температури, рівня води, керування пристроєм та виводу інформації. Вона повністю автоматизована і працює в режимі реального часу. Також створено мобільний додаток для смартфона на базі операційної системи Android, яким можна керувати пристроєм на відстані. Описано усі можливості як пристрою для керованого нагрівання води, так і розробленого мобільного додатка.

4 СТАРТАП-ПРОЕКТ

4.1 Опис ідеї проекту

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Запуск проекту у масове виробництво для побутових та освітніх потреб	1. Освіта	Можливість студентам дізнатись більше про мікропроцесорну техніку
	2. Побутові прилади	Можливість покупцям користуватися технологічним приладом

Таблиця 4.2 – Визначення характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Запропонований метод	Redmond SkyKettle G210S	Xiaomi Mi Smart Kettle			
1.	Вимірювання рівня води	Дає змогу	Не дає змогу	Не дає змогу			+
2.	Вимірювання температури	Дає змогу	Дає змогу	Дає змогу			+
3.	Вартість	70\$	100\$	120\$		+	

4.2 Технологічний аудит ідеї проекту.

У таблиці 4.3 показано оцінку технологічної здійсненності ідеї проекту та наведено технології, що можуть бути використані для реалізації проекту.

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Контроль над пристроєм на відстані	Вимірювання рівня води	Наявна	Доступна
2		Вимірювання температури	Наявна	Доступна
3		Розробка власних апаратно-програмних рішень	Наявна	При обмеженому бюджеті недоступна

Обрана технологія реалізації ідеї проекту: застосування мікроконтролера для реалізації системи контрольованого керуванням нагріву води.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

У таблиці 4.4 показано попередню характеристику потенційного ринку стартап-проекту.

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	500000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Зацікавлення потенційних клієнтів
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	$500000/210000 = 238\%$

У таблиці 4.5 показано характеристику потенційних клієнтів стартап-проекту.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Зростає доступність технологій «розумного будинку»	Компанії з виробутку побутових приладів	Рівень очікування якості пристроя	Відповідність результату найвищим стандартам якості
2	Технологічність пристроїв побутових призначень	Компанії побутових пристроїв	Кожна з потенційних цільових груп має свої вимоги до побутових пристроїв	Забезпечення функціоналу побутового пристроя в залежності від рівня потреб споживача

У табл. 4.6 показані фактори загроз реалізації стартап-проекту.

Таблиця 4.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Незацікавленість клієнтів	Внаслідок невдалого маркетингу клієнт може не зацікавитись послугами	Внесення додаткових сервісних послуг, демонстрація можливостей
2	Втрата конкуренції	Втрата рангу надійного поставника	Якісне та кількісне нарощування інтенсивності та грамотна цінова політика

У табл.4.7 показано фактори можливостей при реалізації стартап- проекту.

Таблиця 4.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Перехід до домінування на ринку побутових приладів	Зростання попиту	Якісне та кількісне нарощування потужностей
2	Імплементация технологій в існуючі системи нагрівання води	Зростання попиту внаслідок зростання клієнтів	Якісне та кількісне нарощування потужностей

У таблиці 4.8 визначено особливості конкурентного середовища та його вплив на впровадження проекту.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Чиста конкуренція	Використання схожих технологій	Стандартизація на високому рівні
2. Локальний	Відсутність єдиного національного постачальника послуг	Окремий підхід до кожної локальної ділянки
3. Міжгалузева	Відсутня	Відсутня
4. Товарно-видова	Застосування стандартизованих технологій	За необхідності, використання загальноновживаних апаратних та програмних засобів
5. Цінова	Застосування спеціалізованих комплексів, які мають значну ціну	Можливість заощадити за допомогою застосування загальноновживаних апаратних засобів
6. Марочна	Кожна діагностика має бути стандартизованою	Отримання переваги на ринку побутових пристроїв

У таблиці 4.9 показано аналіз конкуренції проекту в галузі за М. Портером

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Постачальники побутових пристроїв	Необхідність пошуку постачальників	Залучення малопопулярних постачальників	Незалежність у прийнятті клієнтських рішень	Надання переваги більш авторитетним апаратним рішенням
Висновки:	Середня	Можливість виходу на ринок є	Постачальники диктують цінову політику на обладнання	Клієнти диктують вимоги до якості	Обмеження існують лише у разі відмови від діагностики

У табл. 4.10 показано фактори конкурентоспроможності та їх обґрунтування.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Раціональніший ціновий показник	Можливість більш раціонально використати ресурси на покращення якості виробництва пристрою
2	Надання сервісних послуг	Сервісна підтримка програмної частини

У табл. 4.11 наведено сильні та слабкі сторони проекту.

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні						
			-3	-2	-1	0	+1	+2	+3
1	Раціональніший ціновий показник	17	+						
2	Надання сервісних послуг	12		+					
3	Періодична діагностика	4				+			
4	Необхідність залучення висококваліфікованих кадрів	7							+

У табл.4.12 наведено SWOT-аналіз стартап-проекту.

Таблиця 4.12. SWOT- аналіз стартап-проекту

Сильні сторони: раціональний ціновий показник, надання сервісних послуг	Слабкі сторони: періодична діагностика, можливості погіршень при захопленні руху
Можливості: Перехід до ексклюзивного застосування нової технології, Імплементація технології в існуючі пристрої нагрівання води	Загрози: Незацікавленість клієнтів, втрата авторитету

Альтернативи ринкового впровадження стартап-проекту наведені у табл. 4.13.

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
-----	----------------------------------------------------------------	--------------------------------	-------------------

1	Укладення договорів з компаніями розробки побутових пристроїв при використанні нового рішення	висока	незначні
2	Використання приладів загального вжитку для підвищення конкурентноспроможності	середня	незначні

Обрана альтернатива - Укладення договорів з компаніями розробки побутових пристроїв при використанні нового рішення

4.4 Розроблення ринкової стратегії проекту

Обґрунтування вибору цільових груп потенційних споживачів наведено у табл. 4.14 [42].

Таблиця 4.14. Вибір цільових груп потенційних споживачів

п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Медійні та кінокомпанії	Середня	Високий	Середня	Висока
2	Аматорські кіностудії та рекламні агенства	Висока	Високий	Середня	Низька

Визначення базової стратегії розвитку наведено у табл. 4.15.

Таблиця 4.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Використання альтернативних технологій та пристроїв	Встановлення нового стандарту якості	Зацікавлення та залучення гігантів у галузі побутових пристроїв	Стратегія диференціації
2	Дешевизна проекту	Рациональніші витрати на обладнання, та послуги	Застосування загальноновживаних апаратних рішень замість спеціалізованих комплексів	Стратегія лідерства по витратах

Визначення базової стратегії конкурентної поведінки наведено у табл.4.16.

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Так	Забирати існуючих та шукати нових	Не буде	Стратегія виклику лідера

Визначення стратегії позиціонування наведено у табл. 4.17.

Таблиця 4.17. Визначення стратегії позиціонування

п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Висока якість послуг	Стратегія диференціації	Новизна, гарант якості, точність дослідження	Якість, надійність, точність
2	Мінімальні витрати	Стратегія лідерства по витратах	Універсальність запропонованого рішення	Дешевизна, універсальність

4.5 Розроблення маркетингової програми стартап-проекту

Ключові переваги концепції потенційного товару наведено у табл. 5.18.

Таблиця 5.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Якість	Висока якість, надійність	Надійність
2	Дешевизна	Раціональне використання коштів, дешевше обладнання	Дешевизна

Визначено три рівні моделі товару. Сутність та складові рівнів товару наведено у табл. 4.19.

Таблиця 4.19. Опис трьох рівнів моделі товару

ні товару	Сутність та складові		
I. Товар за задумом	Якісні послуги, стандартизована якість послуг та обладнання		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1)Вартість обслуговування	1) М	1)Е
	2)Кількість комплектів	2) М	2) Пр
	3)Строк безвідмовної праці	3) М	3)Нд
	4)Технологічна собівартість товару	4) М	4)Тх
	Якість: міжнародні стандарти якості, постійна підтримка обладнання		
	Доставка, встановлення та налаштування		
	Марка: Побутова техніка		
III. Товар із підкріпленням	До продажу – обладнання, встановлення		
	Після продажу – сервісна підтримка		

За рахунок чого потенційний товар буде захищено від копіювання: специфічна методика захоплення та обробка даних.

Визначення меж встановлення ціни на послугу наведено у табл. 4.20.

Таблиця 4.20. Визначення меж встановлення ціни

п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	120 у.о./од.	100 у. о./од	Високий	Н. 50 у.о. – В. 70 у.о. (Товар)

Формування системи збуту послуги наведено у табл. 4.21.

Таблиця 4.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Орієнтована на отримання максимальної якості та функціональності приладу нагрівання води	Поставки якісного, точного та надійного товару	Значна	Договірна система збуту

Концепції маркетингових комунікацій наведено у табл. 4.22.

Таблиця 4.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Зацікавленість в якісному та точному продукті з раціональним використанням ресурсів	Медіа ресурси	Гарантованість якості та стандартизація, політика сервісності	Зацікавити у покращеннях пов'язаних із зростаючою популярністю послуг	Представлення центру створення функціонального пристрою нагрівання води
2	Зацікавленість у великій кількості продукту із дотриманням умов якості	Медіа ресурси	Глибина каналу постачальників, гарант якості	Зацікавити у позитивних сторонах первісності та в глибині каналу постачання	Представлення центру створення розумних побутових пристроїв

Висновки до розділу

1. Комерціалізацію стартап-проекту щодо розвитку та впровадження запропонованого апаратно-програмного рішення для створення системи для контрольованого нагрівання води, можна вважати доцільною. На дану пропозицію на ринку побутових пристроїв присутній попит, наразі він задовольняється товарами заміниками та більш дорогими рішеннями, саме тому важливо зайняти нішу конкурента у якості поставника вигідного продукту, порівнюючи з конкурентами. Рентабельність на ринку послуг насамперед обумовлена заміною повної апаратної залежності на універсальність, що обумовлена використанням не спеціалізованих комплексів, а загальноживаного програмного та апаратного забезпечення.

2. Впровадження є перспективним, адже основними групами клієнтів є компанії з розробки побутових приладів, і після набуття достатньої авторитетності можливе охоплення у масштабах міжнаціональних ринків. Конкурентноспроможність проекту обумовлена меншою ціною на повний продукт та більшою функціональністю пристрою в умовах, коли конкуренти за цим параметром у даних умовах програють. Це вигідно вирізняє запропоноване рішення, власне, і є основним критерієм входження на ринок.

3. Альтернативою впровадження було обрано – пошук альтернативних технологій та пристроїв для побудови систем нагрівання води. Імплементация проекту доцільна, оскільки рентабельність та зацікавленість потенційних груп клієнтів створює досить сприятливі умови для розвитку проекту.

ВИСНОВКИ

1. Реалізовано метод створення пристрою для контрольованого нагрівання води, як елемент Інтернету речей.

2. Розроблено мобільний додаток для віддаленого управління пристроєм. Дана система може бути використана як навчальний елемент для лабораторних робіт з курсу «Основи Мікроелектроніки» і як пристрій для побутових цілей.

3. Проведено дослідження мікропроцесорної техніки для реалізації удосконаленої системи контрольованого нагрівання води.

4. Розглянуто історію мікропроцесорної техніки, зроблено огляд мікроконтролерів Arduino Uno та Arduino Mega для системи контрольованого нагрівання води. Наведено опис периферійних пристроїв, які використовують в цій системі та короткий огляд кожного з них та описано принципи їх дії.

5. Проаналізовано додаткові бібліотеки для Arduino IDE. Наведено принципи усунення брязкоту контактів – програмно та апаратно. Реалізовано та описано метод збільшення цифрових входів для мікроконтролера, який використовує лише один аналоговий вхід.

6. Реалізовано метод збільшення цифрових виходів внаслідок використання зсувного регістру.

7. Запропонована удосконалена система для контрольованого нагрівання води на мікроконтролері Arduino Uno та периферійних пристроях, які використовують для вимірювання значення температури, рівня води, керування пристроєм та виводу інформації. Система повністю автоматизована і працює в режимі реального часу

8. Створено мобільний додаток для смартфона на базі операційної системи Android, яким можна керувати пристроєм на відстані. Описано усі можливості як пристрою для керованого нагрівання води, так і розробленого мобільного додатка.

9. Розроблено стартап-проект просування на ринок побутової техніки, удосконаленої системи контрольованого нагрівання води, основу на мікропроцесорної техніці і є елементом Інтернету речей. Показано доцільність та рентабельність проекту та визначено, що комерціалізація проекту є доцільною.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Osborne, Adam (1980). An Introduction to Microcomputers. Volume 1: Basic Concepts (2nd ed.). Berkeley, California: Osborne-McGraw Hill
2. Krishna Kant Microprocessors And Microcontrollers: Architecture Programming And System Design, PHI Learning Pvt. Ltd., 2007
3. History of the Microprocessor URL: <https://meetingtomorrow.com/content-library/history-of-the-microprocessor>
4. National Semiconductor Corporation History URL: <http://www.fundinguniverse.com/company-histories/national-semiconductor-corporation-history/>
5. Hewlett-Packard history URL: <https://www.siliconvalleyhistorical.org/hewlett-packard-history>
6. RISC — архитектура процессора. История и принципы работы. URL: <http://www.xtechx.ru/c40-visokotehnologichni-spravochnik-hitech-book/risc-architecture-processor/>
7. Що таке мікроконтролер Ардуіно URL: <https://learn.sparkfun.com/tutorials/what-is-an-arduino>
8. Різновид мікроконтролерів Arduino URL: <https://store.arduino.cc/arduino-genuino/most-popular>
9. Протокол обмена STK500 URL: <http://microsin.net/programming/avr/avr068-stk500-communication-protocol.html>
10. Atmel ATmega640 Datasheet URL: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf
11. Потенциометры. Виды и устройство. Работа и особенности URL: <https://electrosam.ru/glavnaja/slabotochnye-seti/oborudovanie/potentsiometry/>
12. Фоторезистор. Принцип работы, характеристики URL: <http://www.joyta.ru/7603-fotorezistor-osnovnaya-informaciya/>
13. DS18B20 – датчик температуры URL: <http://mypractic.ru/ds18b20-datchik-temperature-s-interfejsom-1-wire-opisanie-na-russkom-yazyke.html>

14. Цифрове вимірювання рівня гарячої води на мікропроцесорній платформі Arduino // International Scientific And Practical Conference «Technical Sciences: History, The Present Time, The Future, EU Experience», 2019 р.,-С.69-72
15. Медь - коррозионная стойкость URL: <http://delta-grup.ru/bibliot/41/21.htm>
16. Атмосферное давление // Большая советская энциклопедия : [в 30 т.] / гл. ред. А. М. Прохоров. — 3-е изд. — М., 1969—1978
17. Закон Бойля — Мариотта // Физическая энциклопедия / Гл. ред. А. М. Прохоров. — М/, 1988. — Т. 1. — 704 с.
18. Ir Remote And Receiver On An Arduino URL: <http://www.circuitbasics.com/arduino-ir-remote-receiver-tutorial/>
19. Электромагнитные реле. Виды и работа. Устройство и применение. URL: <https://electrosam.ru/glavnaja/jelektrooborudovanie/rozetki-vykljuchateli/elektromagnitnye-rele/>
20. Твердотельные реле URL: <http://electricalschool.info/spravochnik/apparaty/1450-tverdotelnye-rele.html>
21. H-мост и схема работы для управления двигателями URL: <https://arduino-master.ru/uroki-arduino/shema-raboty-n-mosta-dlya-upravleniya-dvigatelyami/>
22. Драйвера мотора на L298N URL: <https://robotchip.ru/obzor-drayvera-motora-na-l298n/>
23. Servo Motor Control using Arduino URL: <https://circuitdigest.com/microcontroller-projects/arduino-servo-motor-control-code-and-circuit>
24. Восприятие цветов человеком URL: <http://rosdesign.com/design/kolorofdesign.htm>
25. Смешение цветов URL: <https://haircolor.org.ua/koloristika/100-smeshenie-tsvetov.html>
26. Пищалка — пьезодинамик Ардуино URL: <https://arduino-master.ru/uroki-arduino/pishhalka-pe zodinamik-arduino/>

27. Construction and Working Principle of LCD Display URL:
<https://www.elprocus.com/ever-wondered-lcd-works/>
28. Bluetooth URL: http://www.smartphone.ua/w_bluetooth.html
29. Arduino IDE software. URL: <https://www.arduino.cc/en/main/software>
30. Библиотеки Arduino URL: <https://doc.arduino.ua/ru/prog/Libraries>
31. Библиотека LiquidCrystal_I2C URL:
https://wiki.iarduino.ru/page/Working_with_character_LCD_displays
32. Библиотека OneWire URL: <https://xakep.ru/2015/05/10/arduino-digital-temp-wire/>
33. Библиотека ServoSmooth URL: <https://alexgyver.ru/servosmooth/>
34. Библиотека GyverButton URL: <https://alexgyver.ru/gyverbutton/>
35. Интерфейсы последовательной связи URL:
<https://studfile.net/preview/357307/page:23/>
36. Интерфейс SPI URL: <http://s-engineer.ru/interfejs-spi/>
37. Интерфейс I2C URL: http://itt-ltd.com/reference/ref_i2c.html
38. Интерфейс UART URL: <https://voltiq.ru/wiki/uart-interface/>
39. Стягивающие и подтягивающие резисторы URL: <http://funnydiy.ru/1010>
40. Постоянная времени RC-цепи URL: https://www.lcard.ru/lexicon/rc_const
41. Datasheet SN74HC14 URL: <http://www.ti.com/lit/ds/symlink/sn74hc14.pdf>
42. Аналогово-цифровой преобразователь в микроконтроллерах
URL: <http://narodstream.ru/avr-urok-22-izuchaem-acp-chast-1/>
43. Последовательное соединение резисторов URL:
<http://hightolow.ru/resistor3.php>
44. Делитель напряжения на резисторах URL:
<http://www.joyta.ru/7328-delitel-napryazheniya-na-rezistorax-raschet-onlajn/>
45. О битовых операциях URL:
<https://tproger.ru/translations/bitwise-operations/>
46. Стягивающие и подтягивающие резисторы URL: <http://funnydiy.ru/1010>
47. Последовательное и параллельное соединение резисторов URL:
<http://hightolow.ru/resistor3.php>

48. Комбинаторика. Сочетания URL:
<https://mathematics.ru/courses/algebra/content/chapter4/section2/paragraph3>
49. Мультиплексоры и демультиплексоры: схемы, принцип работы URL:
<https://pue8.ru/silovaya-elektronika/908-multipleksory-i-demultipleksory.html>
50. Сдвиговый регистр 74НС595 URL:
<https://cxem.net/arduino/arduino166.php>
51. MIT App Inventor URL: <https://appinventor.mit.edu/>
52. Методичні ре-комендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.

ДОДАТОК А
Лістинг програми

```

#include <LiquidCrystal_I2C.h>
#include <OneWire.h>
LiquidCrystal_I2C lcd(0x3F,16,2);
OneWire ds(12); //жёлтый провод
boolean red_butt, blue_butt, yellow_butt, black_butt; //текущие состояние кнопок
boolean red_flag = 0, blue_flag = 0, yellow_flag = 0, black_flag = 0; //предыдущие состояние
кнопок
unsigned long red_last, blue_last, yellow_last, black_last, last_serial, last_buzzer; //последние
нажатие этих кнопок
int level_p;    // уровень потенциометра для lcd экрана
int level_f;    // уровень света из фоторезистора
int lcd_stat;   // состояние переключателя между потенциометром(0) и
фоторезистором(1)
int brightness; // яркость экрана
int serial_val; // для хранения значений Serial.parseInt
int temp_stat;  // выбор между аналоговой(0) и цифровой(1) установкой температуры
int analog_temp; // установка температуры с потенциометра
int digit_temp;  // установка температуры с телефона
int level_temp;  // значение аналоговой/цифровой установки температуры, в зависимости
от состояния
int digit_stat;  // буфер для отправки on(1) и off(0) на телефон
int condition = 0; // состояние системы
int ketler_stat;
int red, blue, green; //значения ШИМ для RGB
int fake_temp;
int temperature = 0; // Глобальная переменная для хранения значение температуры с датчика
DS18B20
long lastUpdateTime = 0; // Переменная для хранения времени последнего считывания с
датчика
const int TEMP_UPDATE_TIME = 1000; // Определяем периодичность проверок
byte simbol[8] = {B00111,B00101,B00111,B00000,B00000,B00000,B00000,B00000,}; // Символ
градуса

```

```

void setup() {
  Serial.begin(9600);
  Serial.setTimeout(10);
  lcd.init();
  lcd.backlight();// Включаем подсветку дисплея
  pinMode(red_pin, INPUT_PULLUP);
  pinMode(blue_pin, INPUT_PULLUP);
  pinMode(yellow_pin, INPUT_PULLUP);
  pinMode(black_pin, INPUT_PULLUP);
  pinMode(red_led, OUTPUT);
  pinMode(green_led, OUTPUT);
  pinMode(blue_led, OUTPUT);
  pinMode(lcd_pin, OUTPUT);
  pinMode(lcd_knopka, INPUT);
  pinMode(poten_lcd, INPUT);
  pinMode(poten_temp, INPUT);
  pinMode(fotor_pin, INPUT);
  pinMode(chainik, INPUT);
  digitalWrite(rele, HIGH); //на всякий случай выключаю реле (чайник)
}

void loop() {
  if (temperature < 5) ketler_stat = 1;
  else ketler_stat = 1;
  //КОД ДЛЯ ВЫБОРА между автояркостью LCD ЭКРАНА и ручной настройки
  lcd_stat = digitalRead(lcd_knopka);
  level_p = analogRead(poten_lcd);
  level_f = analogRead(fotor_pin);
  level_p = map (level_p, 0, 1024, 0, 255);
  level_f = map (level_f, 500, 1024, 255, 0);
  level_f = constrain(level_f, 0, 255);
  switch (lcd_stat) {
    case 0: analogWrite(lcd_pin, level_p); brightness = map (level_p, 0, 255, 0, 100);
      break;

```



```

    case 1: analogWrite(lcd_pin,level_f); brightness = map (level_f, 0, 255, 0, 100);
        break;    }
//РАБОТА С ТЕМПЕРАТУРАМИ
switch(temp_stat){
    case 0: level_temp = analog_temp;
        break;
    case 1: level_temp = digit_temp;
        break; }
analog_temp = analogRead(poten_temp);
analog_temp = map (analog_temp,5,1024,0,100);
detectTemperature(); // Определяем температуру от датчика DS18B20
// fake_temp = analogRead(chainik);
// temperature = map (fake_temp,0,1024,0,165);
//КОД для приёма данных по блютуз
if (Serial.available()>0)
{
    serial_val = Serial.parseInt();
    if (serial_val >= 0 && serial_val <= 100)
    { digit_temp = serial_val; }
    else if (serial_val == 301)
    { condition = 1;}
    else if (serial_val == 300)
    { condition = 0;}
    else if (serial_val == 302)
    { condition = 2;}
    else if (serial_val == 303)
    { condition = 3;}
    else if (serial_val == 304)
    { condition = 4;}
    else if (serial_val == 305)
    { condition = 5;}
}
//КОД ДЛЯ ЧТЕНИЯ КНОПОК

```

```

red_butt = !digitalRead(red_pin); // считать текущее положение кнопки
blue_butt = !digitalRead(blue_pin);
yellow_butt = !digitalRead(yellow_pin);
black_butt = !digitalRead(black_pin)
//вкл
if (red_butt == 1 && red_flag == 0 && millis() - red_last > 100) {
    red_flag = 1;
    condition = 1;
    red_last = millis();
    if (red_butt == 0 && red_flag == 1) {
        red_flag = 0; }
//аналоговая ПОСТОЯННАЯ ПОДДЕРЖКА
if (blue_butt == 1 && blue_flag == 0 && millis() - blue_last > 100) {
    blue_flag = 1;
    temp_stat = 0;
    condition = 3;
    blue_last = millis(); }
if (blue_butt == 0 && blue_flag == 1) {
    blue_flag = 0; }
//аналоговая УСТАНОВКА ОДИН РАЗ
if (yellow_butt == 1 && yellow_flag == 0 && millis() - yellow_last > 100) {
    yellow_flag = 1;
    temp_stat = 0;
    condition = 2;
    yellow_last = millis(); }
if (yellow_butt == 0 && yellow_flag == 1) {
    yellow_flag = 0; }
//выкл
if (black_butt == 1 && black_flag == 0 && millis() - black_last > 100) {
    black_flag = 1;
    condition = 0;
    black_last = millis(); }
if (black_butt == 0 && black_flag == 1) {

```

```

    black_flag = 0;
//КОД для выбора состояния чайника (основная часть)
switch (ketler_stat) {
case 1
switch (condition) {
case 0: digitalWrite(rele, HIGH); lcd.setCursor(0, 1); lcd.print("stat = OFF ");
temp_stat = 0; digit_stat = 0;
break;
case 1:
if (temperature < 97 ) {
digitalWrite(rele, LOW); lcd.setCursor(0, 1); lcd.print("stat = ON ");
digit_stat = 1;
}
else if (temperature = 98) {
condition = 0; digitalWrite(rele, HIGH);
digitalWrite(red_led, HIGH); digitalWrite(blue_led, HIGH); digitalWrite(green_led, HIGH);

else if (temperature > 98) {
condition = 0;
}
break
case 2:
if (temperature < analog_temp ) {
digitalWrite(rele, LOW); lcd.setCursor(0, 1); lcd.print("stat = ~ON~");
digit_stat = 1;
} else {
condition = 0;
}
break;
case 3:
if (temperature < analog_temp ) {
digitalWrite(rele, LOW); lcd.setCursor(0, 1); lcd.print("stat = HOLD");
digit_stat = 1;

```

```

} else {
    digitalWrite(rele, HIGH); lcd.setCursor(0, 1); lcd.print("stat = WAIT");
    digit_stat = 2;
}
break;
case 4:
    if (temperature < digit_temp ) {
        digitalWrite(rele, LOW); lcd.setCursor(0, 1); lcd.print("stat = d.ON");
        temp_stat = 1; digit_stat = 1;
    } else {
        condition = 0; temp_stat = 0; digit_stat = 0;
    }
    break;
case 5:
    if (temperature < digit_temp ) {
        digitalWrite(rele, LOW); lcd.setCursor(0, 1); lcd.print("stat=d.HOLD");
        temp_stat = 1; digit_stat = 1;
    } else {
        digitalWrite(rele, HIGH); lcd.setCursor(0, 1); lcd.print("stat=d.WAIT");
        digit_stat = 2;
    }
    break;
}
//КОД ДЛЯ RGB СВЕТОДИОДА
if (temperature < 30 )           { red = 0; blue = 100; green = 0; } //синий
if (temperature > 30 && temperature < 50) { red = 135; blue = 85; green = 0; } //голубой
if (temperature > 50 && temperature < 70) { red = 255; blue = 0; green = 55; } //жёлтый
if (temperature > 70 && temperature < 90) { red = 255; blue = 0; green = 15; } //оранжевый
if (temperature > 90)           { red = 255; blue = 0; green = 0; } //красный
switch (digit_stat){
    case 0: digitalWrite(red_led, LOW); digitalWrite(blue_led, LOW); digitalWrite(green_led, LOW);
        break;
    case 1: analogWrite(red_led, red); analogWrite(green_led, green); analogWrite(blue_led, blue);

```

```

    break
case 2: digitalWrite(red_led, LOW); digitalWrite(blue_led, LOW);
if (round(millis() / 700) % 2 == 0)
    analogWrite(green_led, 80);
else digitalWrite(green_led, LOW);
    break; }

//КОД ДЛЯ ОТОБРАЖЕНИЯ НА LCD ЭКРАНЕ
lcd.createChar(1, simvol);
lcd.setCursor(0, 0); lcd.print("temp =");
if (temperature > 99){ lcd.setCursor(6, 0); lcd.print(temperature); }
else if (temperature < 10) { lcd.setCursor(6, 0); lcd.print(" "); lcd.print(temperature); }
else { lcd.setCursor(6, 0); lcd.print(" "); lcd.print(temperature); }
lcd.setCursor(9,0); lcd.print("\1");
lcd.setCursor(10,0); lcd.print("C|"); //ТЕКУЩАЯ ТЕМПЕРАТУРА
if (level_temp < 10 ){ lcd.setCursor(12, 0); lcd.print(" "); lcd.print(level_temp); }
else {lcd.setCursor(12, 0); lcd.print(level_temp);}
lcd.setCursor(14,0); lcd.print("\1"); lcd.print("C"); //УСТАНОВЛЕННАЯ ТЕМПЕРАТУРА

lcd.setCursor(11, 1); lcd.print("|"); //чёрточка после СТАТУС СИСТЕМЫ
if (brightness > 99 ) { lcd.setCursor(12, 1); lcd.print(brightness); }
else if (brightness < 10) { lcd.setCursor(12, 1); lcd.print(brightness); lcd.setCursor(13, 1); lcd.print("
"); }
else { lcd.setCursor(12, 1); lcd.print(brightness); lcd.setCursor(14, 1); lcd.print(" "); }
lcd.setCursor(15, 1); lcd.print("%"); //ЯРКОСТЬ ПОДСВЕТКИ ЭКРАНА
    break; //брэйк для свитч кетлет стат кэуз 1
analogWrite(red_led, 50); analogWrite(green_led, 0); analogWrite(blue_led, 20);
break;

//КОД ДЛЯ ВЫВОДА В SERIAL PORT
if (millis() - last_serial > 500) {
    Serial.print(temperature); Serial.print(" C"); Serial.print("|");
    Serial.print(level_temp); Serial.print("|");
    switch (digit_stat) {

```

```

case 0: Serial.print("OFF"); Serial.println();
    break;
case 1: Serial.print("ON"); Serial.println();
    break;
case 2: Serial.print("WAIT"); Serial.println();
    break;
}
last_serial = millis();
}
}

int detectTemperature(){
    byte data[2]; // Место для значения температуры
    ds.reset(); // Начинаем взаимодействие со сброса всех предыдущих команд и параметров
    ds.write(0xCC); // Даем датчику DS18B20 команду пропустить поиск по адресу. В нашем
случае только одно устройство
    ds.write(0x44); // Даем команду измерить температуру.
    // Само значение температуры мы еще не получаем. Датчик его положит во внутреннюю
память

    if (millis() - lastUpdateTime > TEMP_UPDATE_TIME)
    {
        lastUpdateTime = millis();
        ds.reset(); // Теперь готовимся получить значение измеренной температуры
        ds.write(0xCC);
        ds.write(0xBE); // Просим передать нам значение регистров со значением температуры
        data[0] = ds.read(); // Читаем младший байт значения температуры
        data[1] = ds.read(); // А теперь старший

        // Формируем значение
        temperature = (data[1] << 8) + data[0];
        temperature = temperature >> 4;
        temperature = temperature + 2;
    }
}

```

ДОДАТОК Б
ABSTRACT

The design of the device begins with the identification of all the necessary peripheral devices that are used to implement a system of controlled heating of water, namely:

- potentiometer;
- photoresistor;
- DS18B20 temperature sensor;
- MPXV5050 pressure sensor;
- infrared module;
- solid state relay;
- engine driver;
- servomotor;
- water pump;
- RGB - LED;
- sound module;
- LCD 2004 display;
- Bluetooth module;
- clock buttons;
- switches;
- wires to connect.

And you also need an Arduino Uno microcontroller to control all the peripherals. All of the above components are shown in Figure 1.



Figure 1 - Components for device design

Schematic diagram of the electrical device

The peripherals are connected to the microcontroller according to the schematic diagram shown in Figure 2.



Figure 2 - Schematic diagram of the principle

Connection of electrical components

Before you can solder electrical components, you need to connect them to the breadboard to make sure the device is working properly. An example of pre-connecting the clock buttons and potentiometers to the microcontroller on the motherboard is shown in Figure 3.

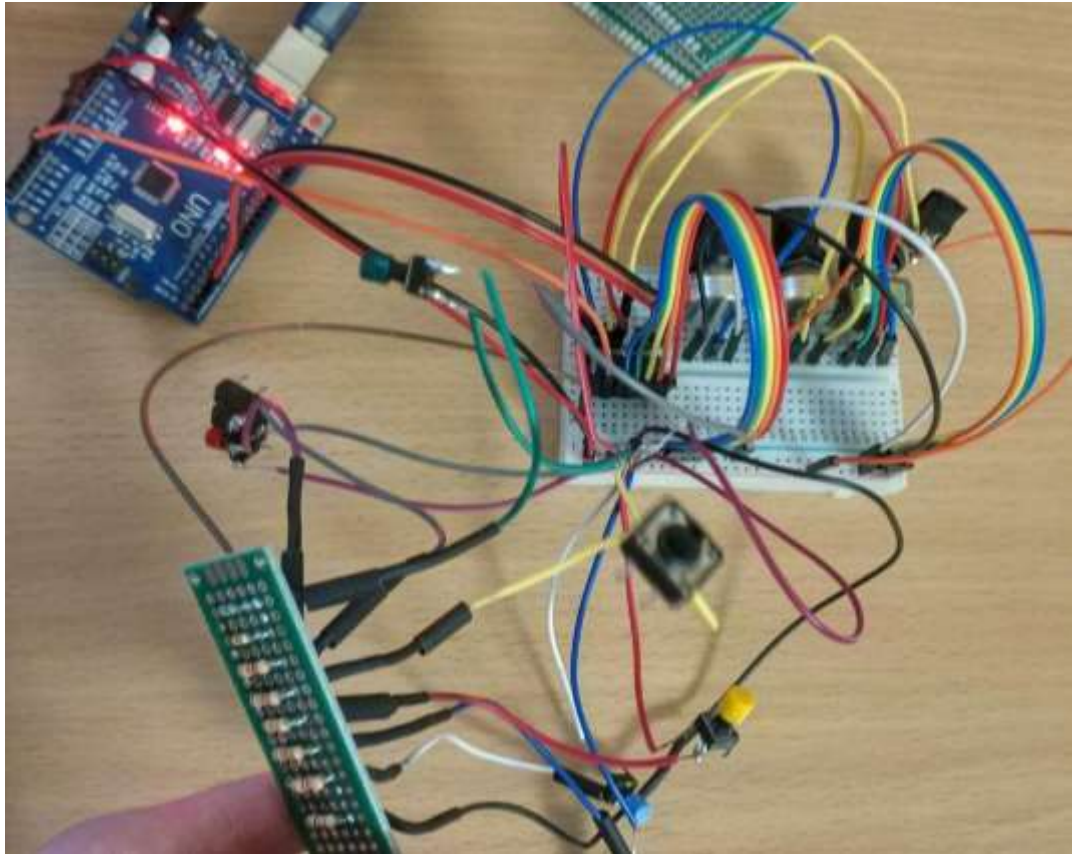


Figure 3 - Connection of components on a breadboard

After checking the functionality of the pre-connection of electrical components, we connect all other peripherals with each other. A prerequisite is to combine the earth of each component into a common wire. The Earth synchronizes the peripherals with each other and ensures the correct operation of the system. The connections of all components are shown in Figure 4.

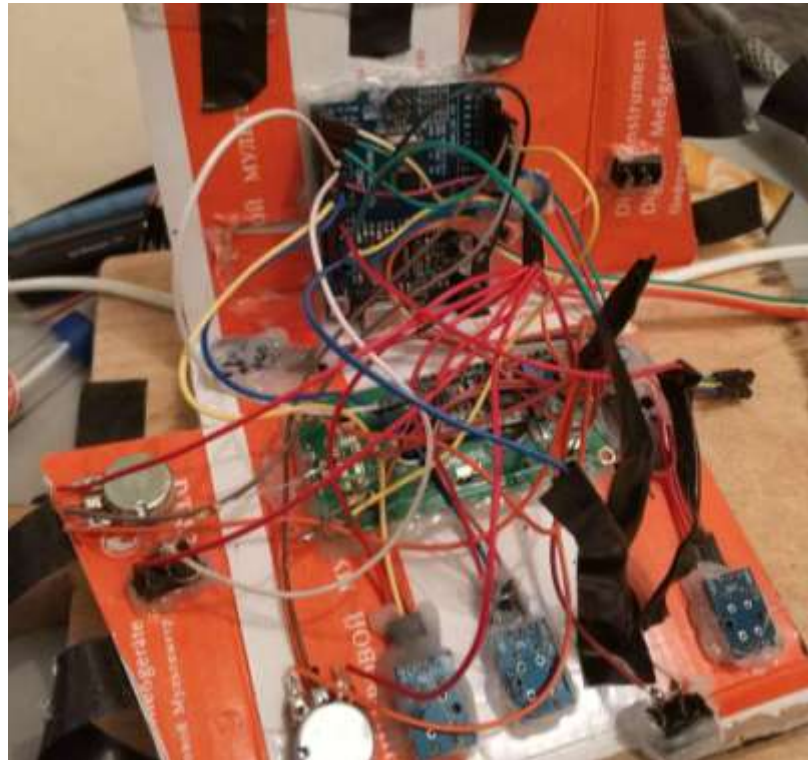


Figure 4 - Connect all electrical components to each other

Layout for the device body

The housing layout for the controlled water heating system is made of PVC paper and is painted in black. All parts of the layout are bonded with thermoplastic glue. The layout has the following options:

- width –127 mm;
- length - 105 mm;
- height - 35 mm;
- weight - 192 g.

The outer part of the layout contains the following elements:

- clock buttons;
- potentiometers;
- switches;
- photoresistor;
- sound module;
- power outlet;
- output for firmware download;

- RGB LED.

The layout of the device is shown in Figure 5



Figure 5 - Appearance layout of the device

An example of software implementation

A C ++ program code was developed for the device to function, which was downloaded to the Arduino Uno microcontroller via an Arduino IDE compiler, developed by the same manufacturer - Arduino.ss.

The program code developed is given in Appendix A.

Validation of code for performance in the Arduino IDE compiler:

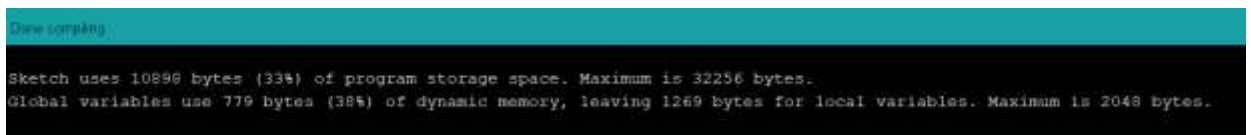


Figure 6 - Arduino IDE Compiler Console

The principle of operation of the device

The implemented device is able to measure in real time the water level in 100 ml increments and the temperature in 1 ° C increments. Therefore, first of all, there is a security system that prohibits the device from operating when the kettle or the kettle is less than the minimum level connected to the system. Using the ability to measure the level and temperature, the device has the following modes:

7.) Water heating to 100 ° C

When the Start button is pressed, the microcontroller instructs the relay to close and the temperature sensor measures the state of water heating. When the temperature reaches 100 ° C, the relay automatically unlocks, thus stopping the water heating process.

8.) Water heating to the set temperature.

To set the temperature, use the potentiometer knob to set the temperature to which the water is to be heated. Further, when the Set Temp button is pressed, the device performs all operations as in the first mode, but the heating process will stop when the temperature sensor locks the set value on the potentiometer. During this mode, the potentiometer value can be adjusted to change the set temperature.

9.) Keeping the water temperature at the set point.

To keep the temperature at the setpoint, you must use the same potentiometer as in the second mode. After setting the temperature value, press the "Hold Temp" clock button to start the temperature holding mode. When the set temperature is reached, the heating process stops until the water temperature is lower than that set on the potentiometer. You can deactivate this mode by pressing the Stop button or switching on another mode.

10.) Setting the water level to the setpoint.

Another potentiometer, which is responsible for the value of the water level, set the required volume of water that should be contained in the water heater. With the "Set Water" button, we start the process of collecting water. This process stops when the water level reaches the setpoint. Water installation is only possible when the unit is present in the system and does not heat the water.

11.) Keeping the water level at the set value.

To maintain the water level, it is necessary to set the level value on the same potentiometer used in the fourth mode. When the Hold Water clock button is pressed, the unit checks the presence of the device in the system and checks for no water heating at this point in time. If these conditions are met and the water level in the unit is less than the set level, the water pump pumps the water to the setpoint.

12.) Emergency stop mode

Pressing the Stop button stops any of the above modes. This mode is used to turn off the temperature and water level hold modes, as well as to stop the water from heating up or to get water to the unit.

Other technical features of the device:

- You can choose to automatically or manually control the brightness of the LCD. When selecting manual mode, you can adjust the brightness with a separate potentiometer. When selecting automatic, the brightness changes according to the light of the room where the system is installed.
- The sound module generates messages about the end of water heating and the setting of the appropriate water level in the system, as well as some other system signals.
- The RGB LED informs you of the current system mode and changes its color from cold blue to warm red according to the water temperature during heating.

Developing a software environment on a smartphone to control the device

Controlled water heating can be controlled via a mobile application on an Android smartphone [51]. App Inventor development environment was used to create the mobile application.

The MIT App Inventor 2 development environment

MIT App Inventor is an intuitive visual programming environment that allows you to create fully functional applications for smartphones and tablets [52]. Block-based App Inventor 2 helps build complex applications in much less time than traditional programming environments. The MIT App Inventor aims to democratize software development by enabling all people to move from technology consumption to technology creation.

App Inventor uses graphical user interface for programming, the visual programming language is very similar to Scratch and StarLogo TNG. A compiler that translates the App Inventor visual block language into an Android byte code based on the GNU framework for implementing Kawa dynamic languages, which implements

Scheme for the Java (Android) platform. This software product was developed by Google using the Open Blocks Java library developed by MIT

A small team of CSAIL staff and students, led by Professor Hal Abelson, support this free online application development environment, which is used by over 6 million registered users

An example of software implementation

The mobile app graphic design was originally created in the Designer section. The App Inventor 2 environment interface is shown in Figure 7.

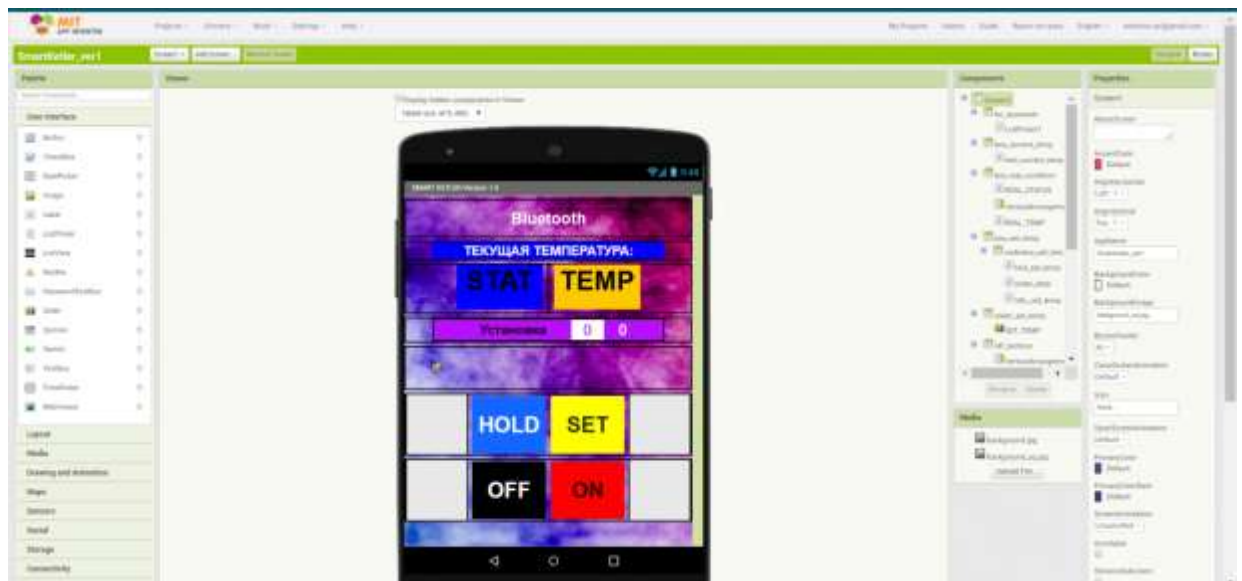


Figure 7 - App Inventor 2 development environment

The code is created in the Blocks section, which uses block diagrams to program individual graphic design elements. An example of a block code for an application that can be controlled by a developed system of controlled water heating is shown in Fig. 8.

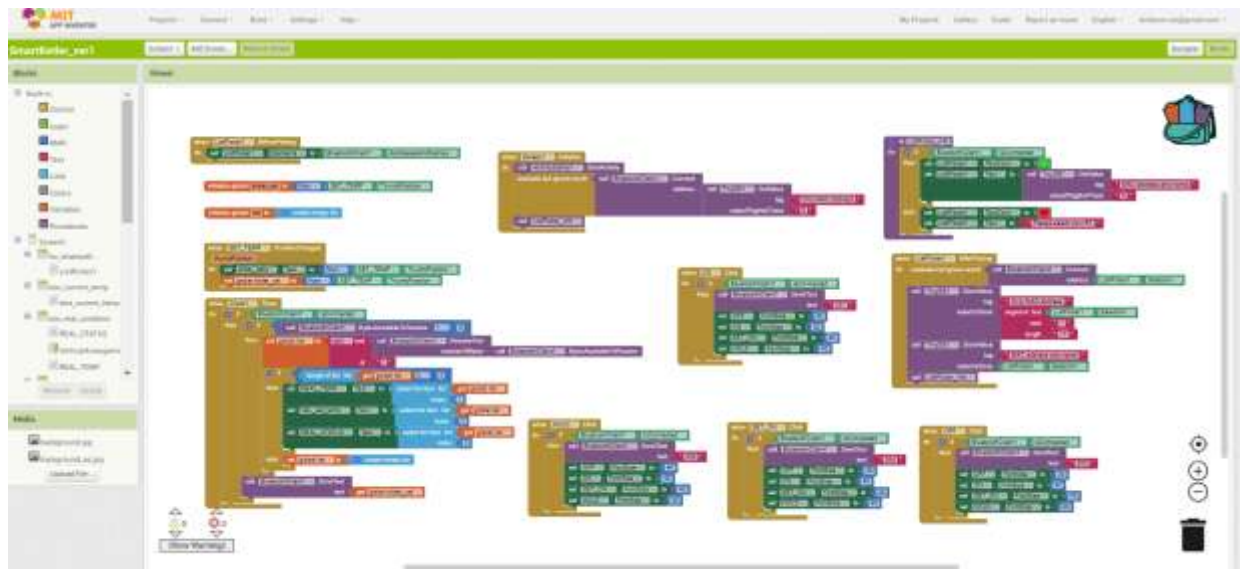


Figure 8 - Mobile application code from blocks

To test the functionality of the application, we create an APK file that we download to a smartphone and connect to the Bluetooth module of a developed device of the water heating system. Screen capture from a smartphone with the validation of the mobile application is shown in Figure 9.



Figure 9 - Mobile application for device management

The principle of operation of the program

The principle of operation of the mobile application completely duplicates the control of the device with the clock buttons. All information is exchanged via Bluetooth. When you open a mobile application, it automatically searches for the device's pre-set IP address and connects automatically.. The buttons in the application trigger the respective modes of the device, namely:

- “Start” - Water heating to 100 ° C;
- “Set Temp” - Water heating to the set temperature;
- "Hold Temp" - Keep the water temperature at the set point;
- “Set Water” - Setting the water level to the set value;
- “Hold Water” - Keeping the water level set;
- "Stop" - Emergency stop mode.